Code Protection **through Obfuscation**

Pedro Fortuna

SecAppDev 2018

jscrambler

About Me

PEDRO FORTUNA

CO-FOUNDER & CTO @ **JSCRAMBLER**

SECURITY, JAVASCRIPT
@PEDROFORTUNA

Agenda

1
What is Code Obfuscation?

2
Obfuscation Concepts

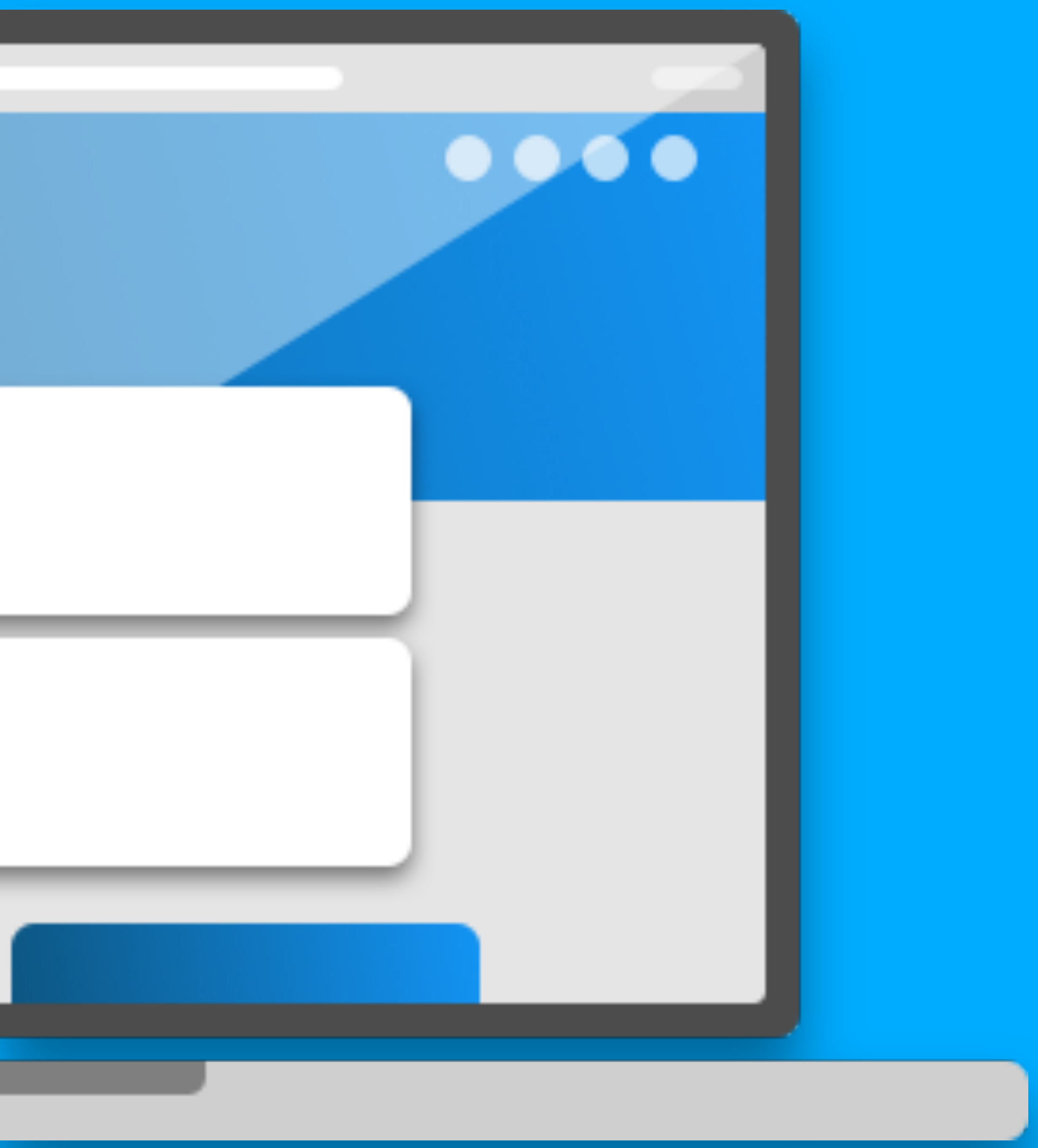
3
Obfuscation Process

4
Obfuscation Transformations

5
Beyond Obfuscation

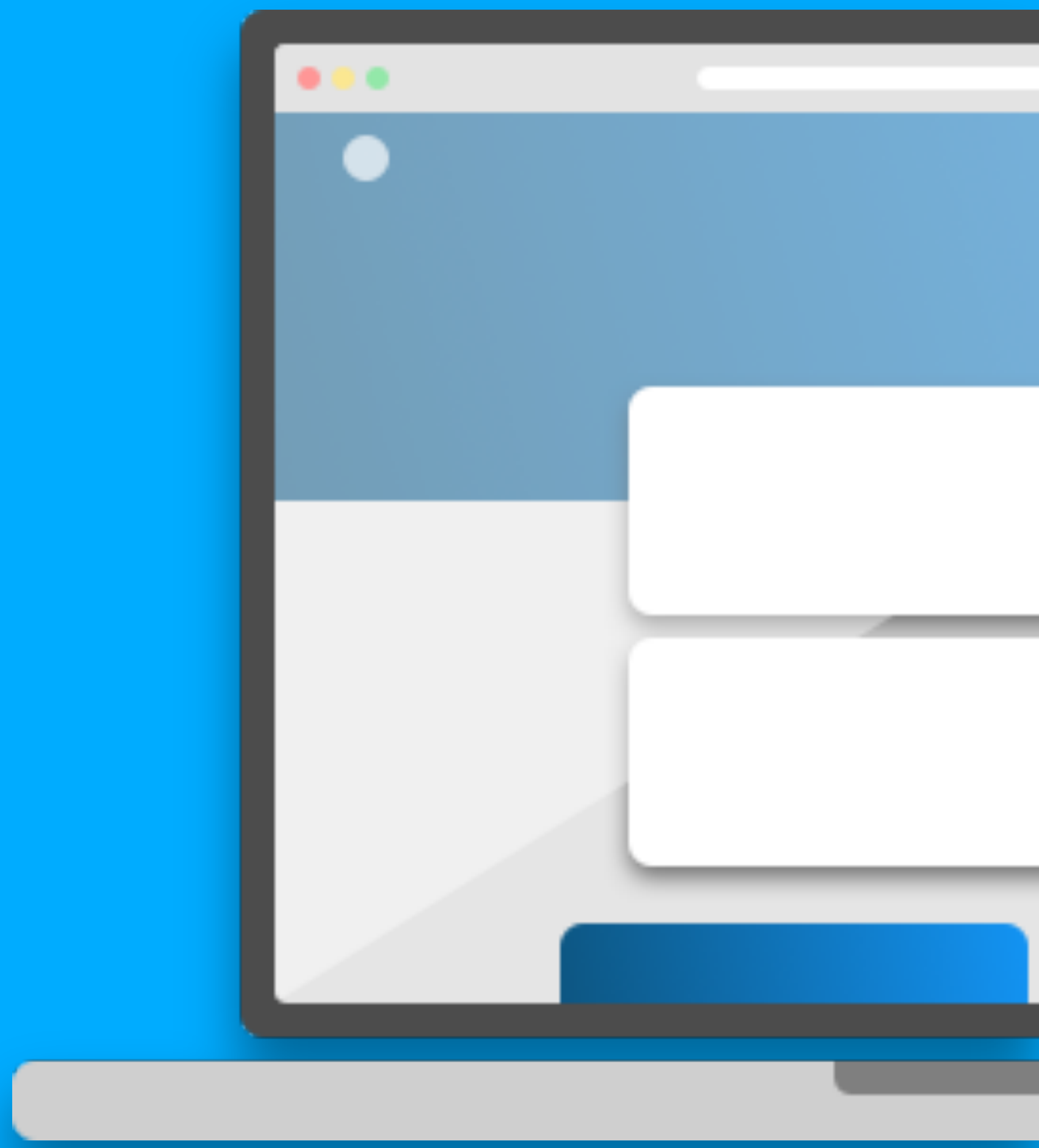
6
Conclusions

7
Q & A



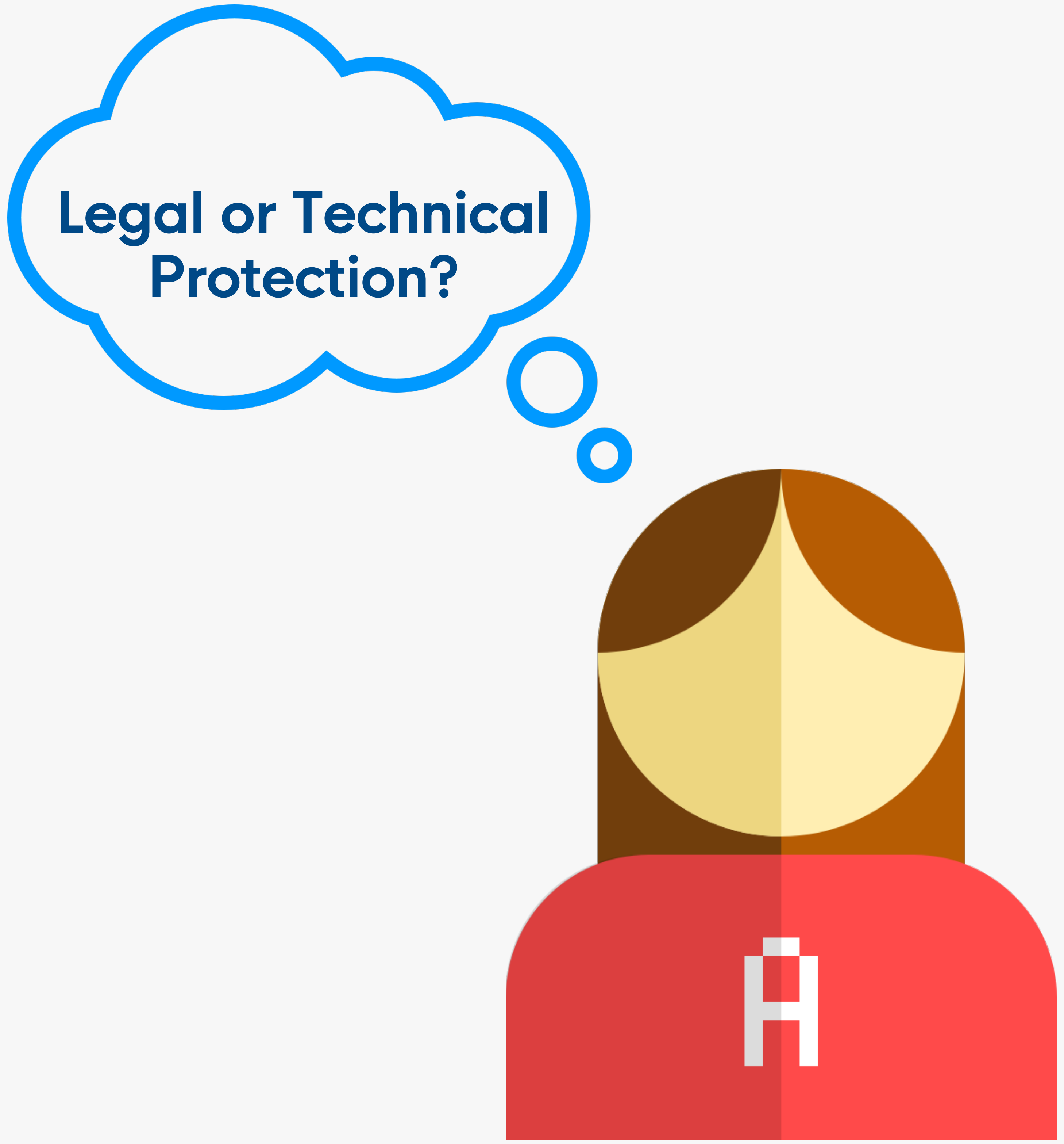
WHAT IS CODE OBFUSCATION

PART 1



jscrambler

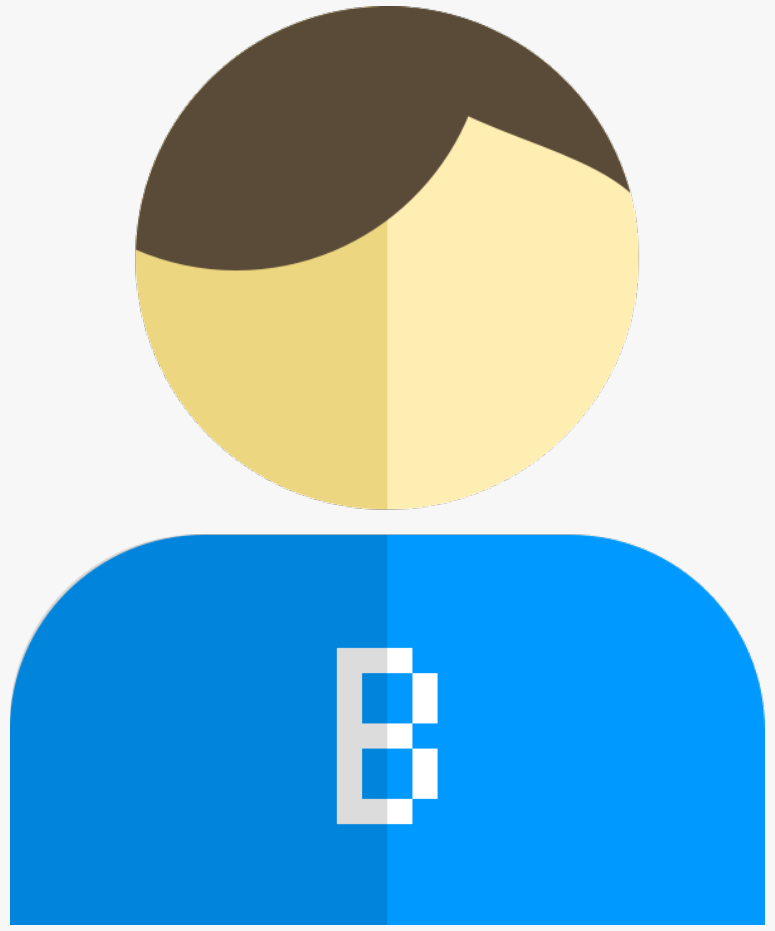
Intellectual Property **Protection**



Alice

Software Developer
Sells her software over the Internet

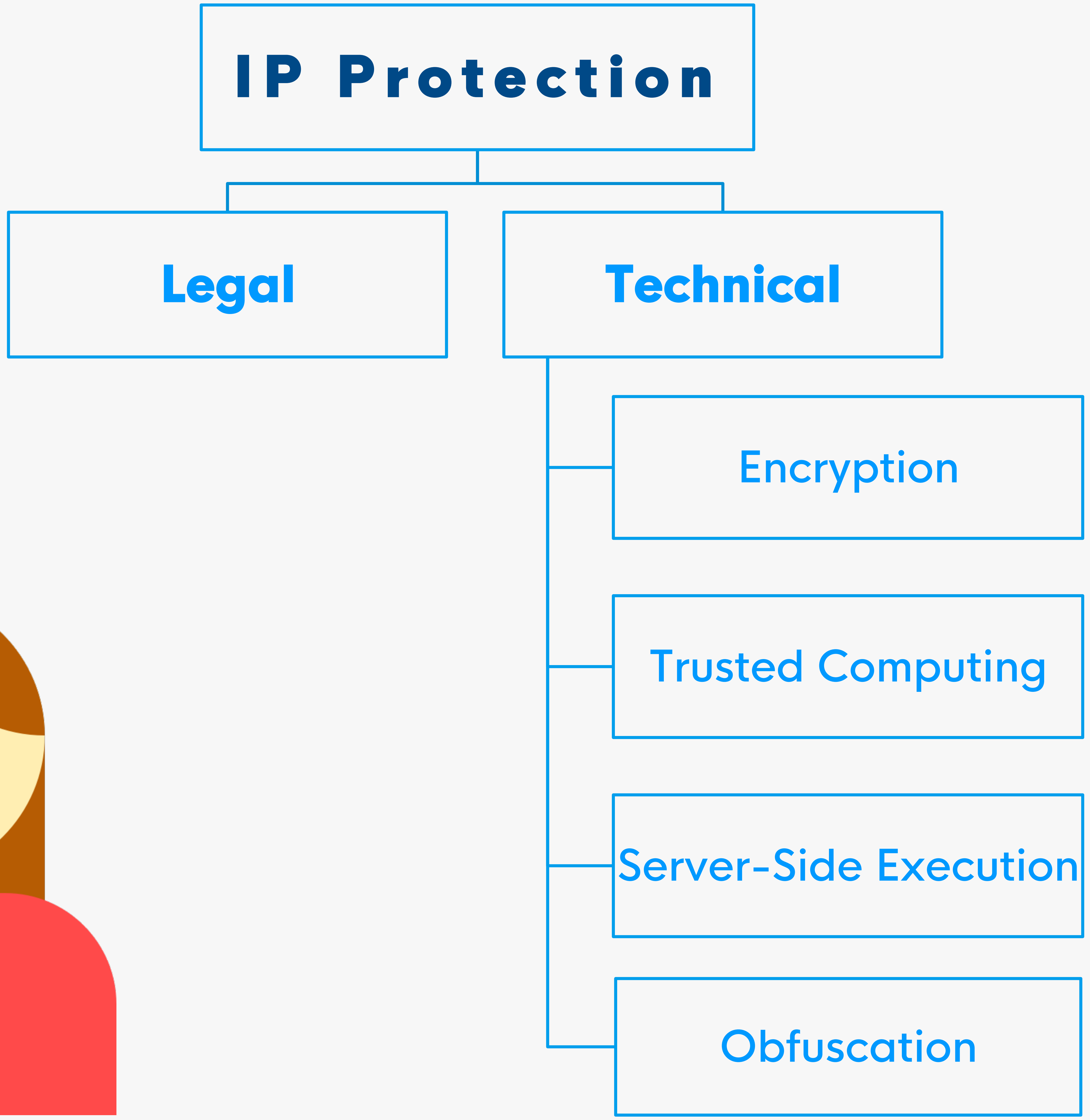
VS



Bob

Reverse Engineer
Wants key algorithms and data structures
Does not need to revert back to original source code

Intellectual Property **Protection**



- IP Protection
  - Legal
  - Technical
    - Encryption
    - Trusted Computing
    - Server-Side Execution
    - Obfuscation

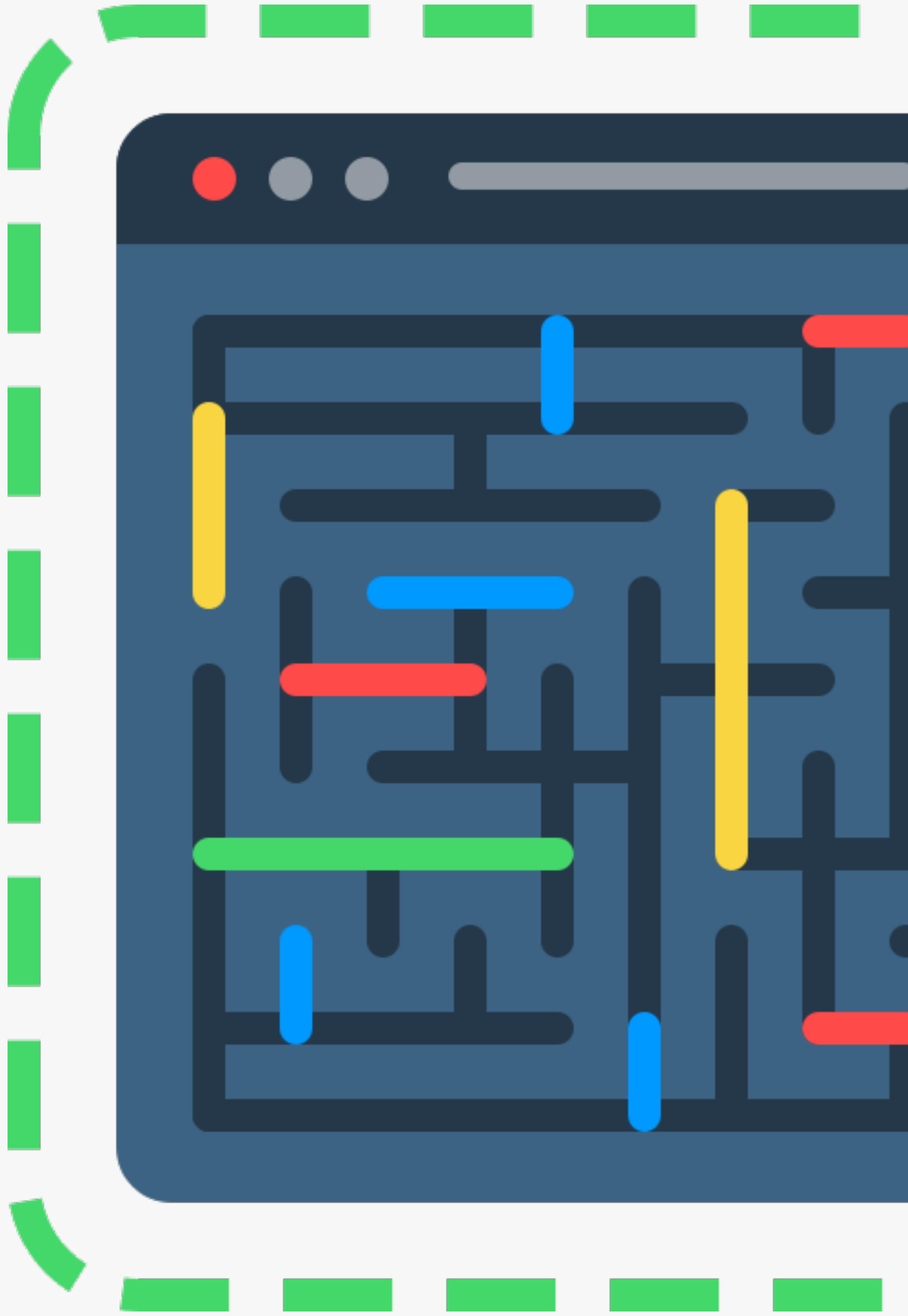
So when does **it make sense?**

- **When offering the sensitive computation on the server is not an option**
    - **You may not have one**
        - Standalone offline playable games
        - Mobile applications
        - Widgets / UI Controls
        - Desktop applications (Electron, NW.js)
    - **You may not want one**
        - May not be cost effective doing computations on a server (you have to guarantee 100% uptime, support teams)
        - Latency

So when does **it make sense?**

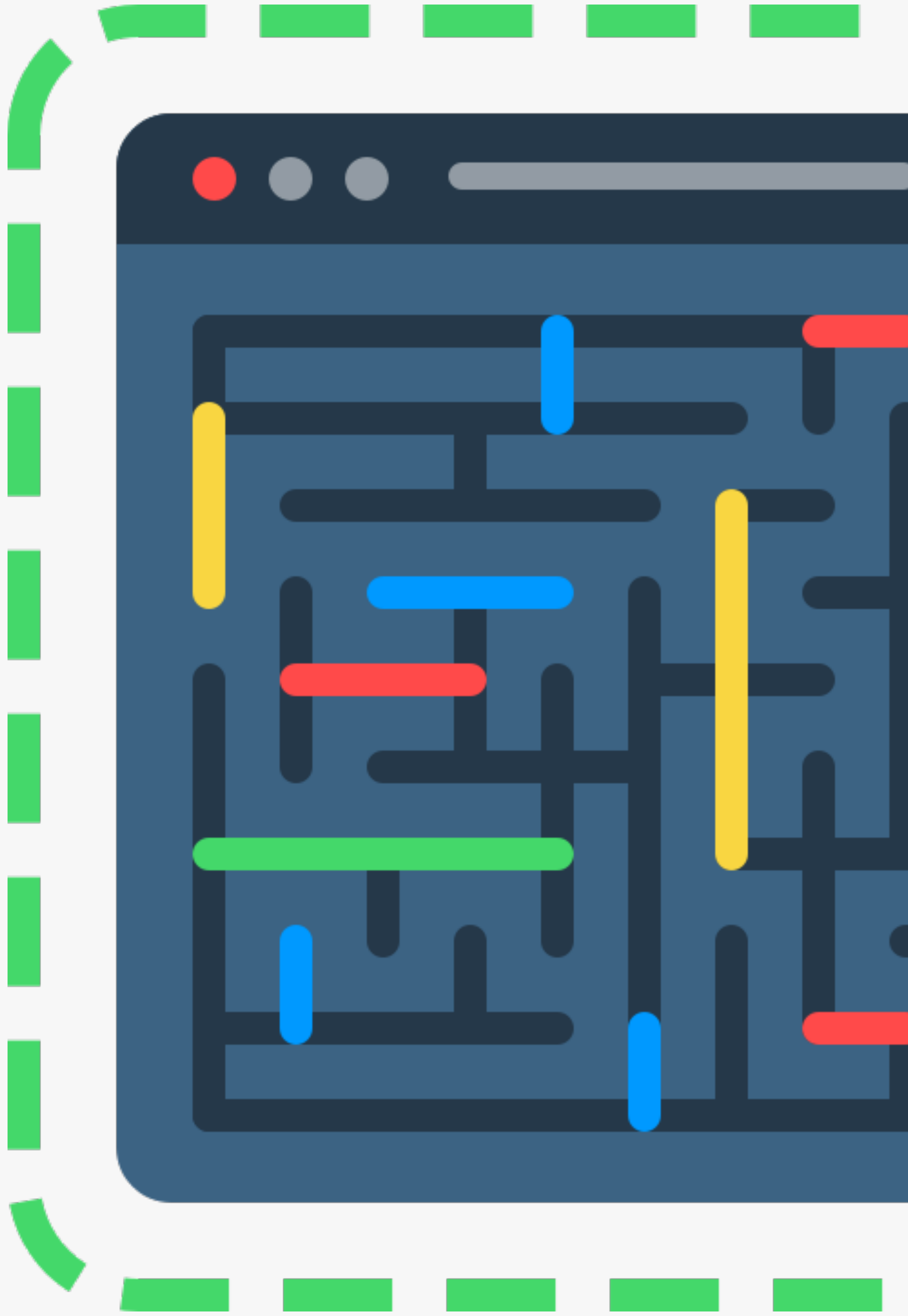
- **When using trusted computing is not an option**
  - Not generally available in everyday devices => reduced clientele
  - Cost

- **When adversaries have physical access to the system and to the code (Man At The End - MATE)**
  - (some) Mobile applications
  - IoT
  - Gadgets
  - Desktop applications
  - On prem deployments
  - A growing number of Web Applications



So when does **it make sense?**

- **Web applications are being target by bots**
  - Crawlers
  - Automated account registration
  - Abuse
  - Malicious extensions
  - UI Redressing / Clickjacking
  - Cryptojacking
  - Man in the Browser (MITB) attacks
  - …



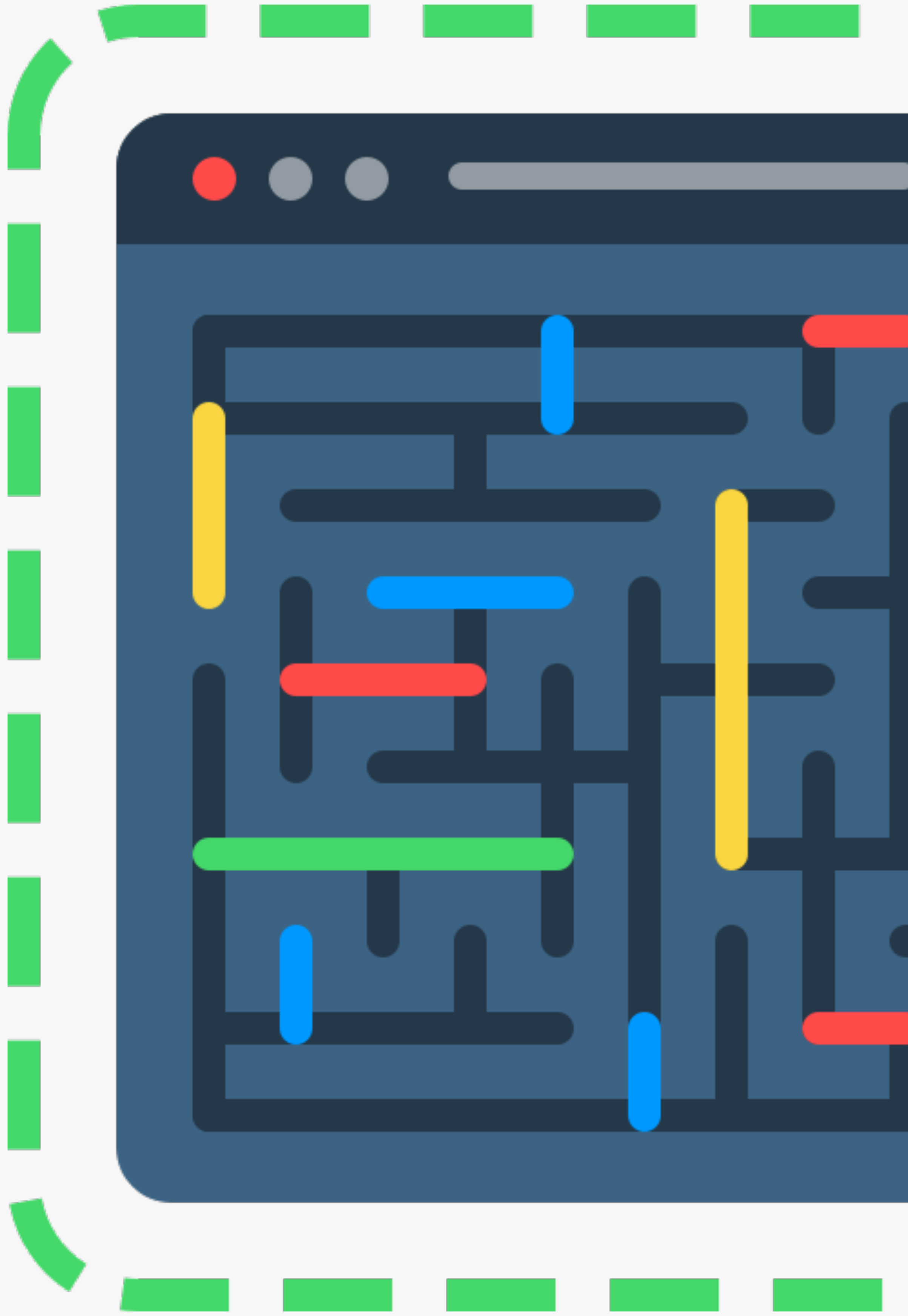
Code **Obfuscation**

Obfuscation

“transforms a program into a form that is **more difficult** for an adversary to understand or change than the original code” [1]

More Difficult

“requires more human time, more money, or more computing power to analyze than the original program.”



[1] in Collberg, C., and Nagra, J., “Surreptitious software: obfuscation, watermarking, and tamperproofing for software protection.”, Addison-Wesley Professional, 2010.

Code **Obfuscation**

Lowers the Code Quality in terms of

Readability

Delay program understanding

Time required to reverse it > program useful lifetime

Resources needed to reverse it > value obtained from reversing it

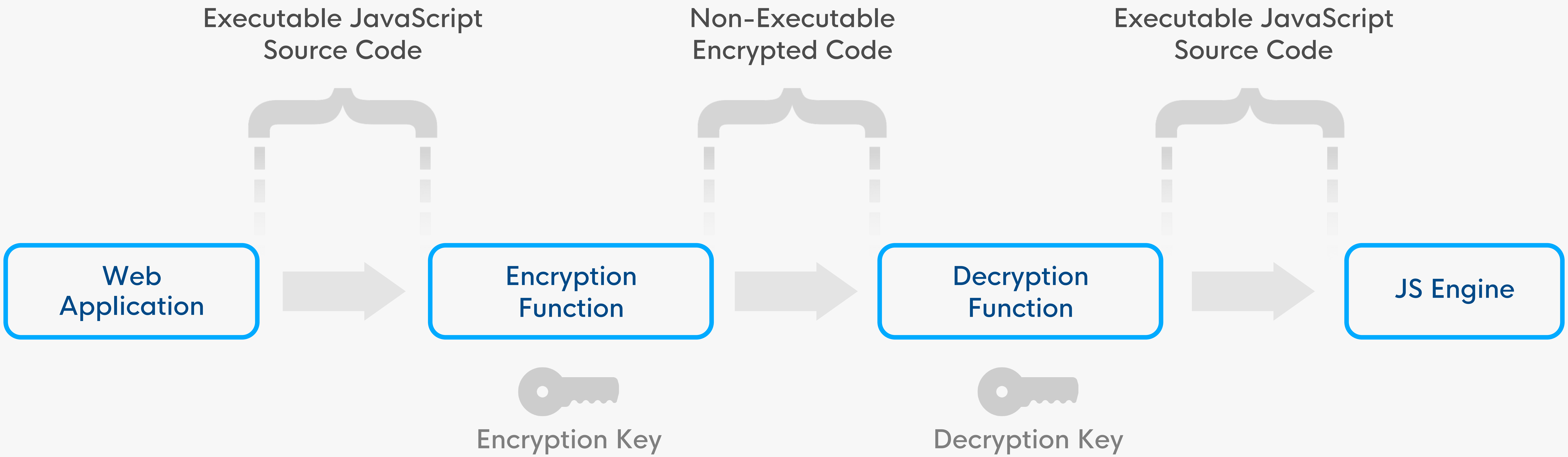
Maintainability

Delay program modification

Cost reversing it > cost of developing it from scratch

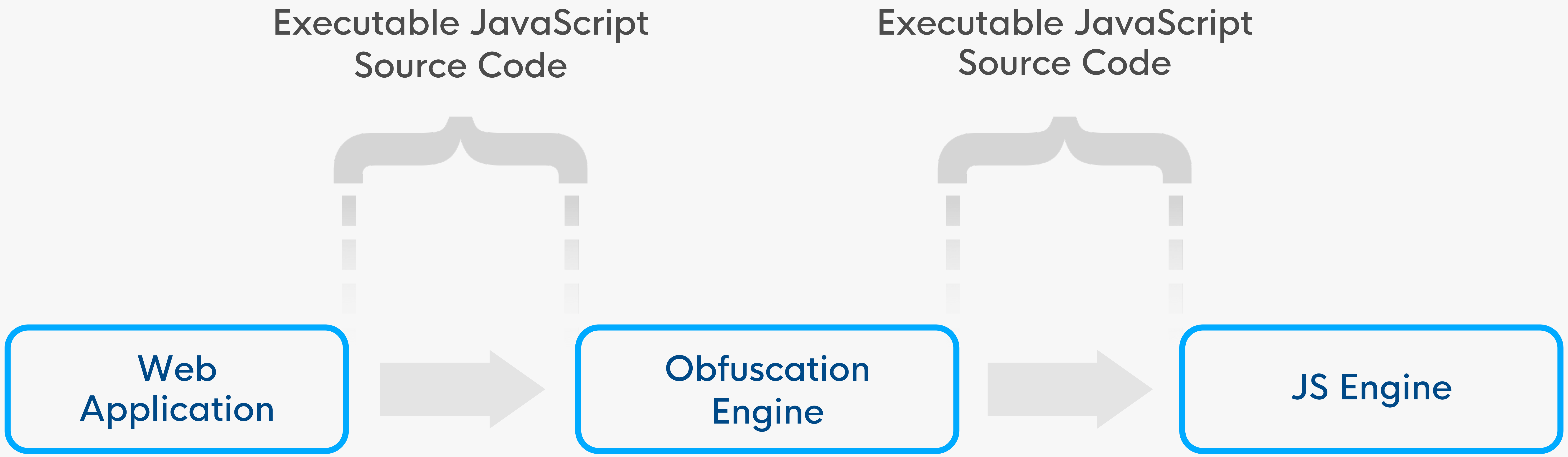
Manually reversing obfuscation is **always** possible

Code Encryption vs **Obfuscation**


Executable JavaScript Source Code
Non-Executable Encrypted Code
Executable JavaScript Source Code
Web Application
Encryption Function
Decryption Function
JS Engine
Encryption Key
Decryption Key

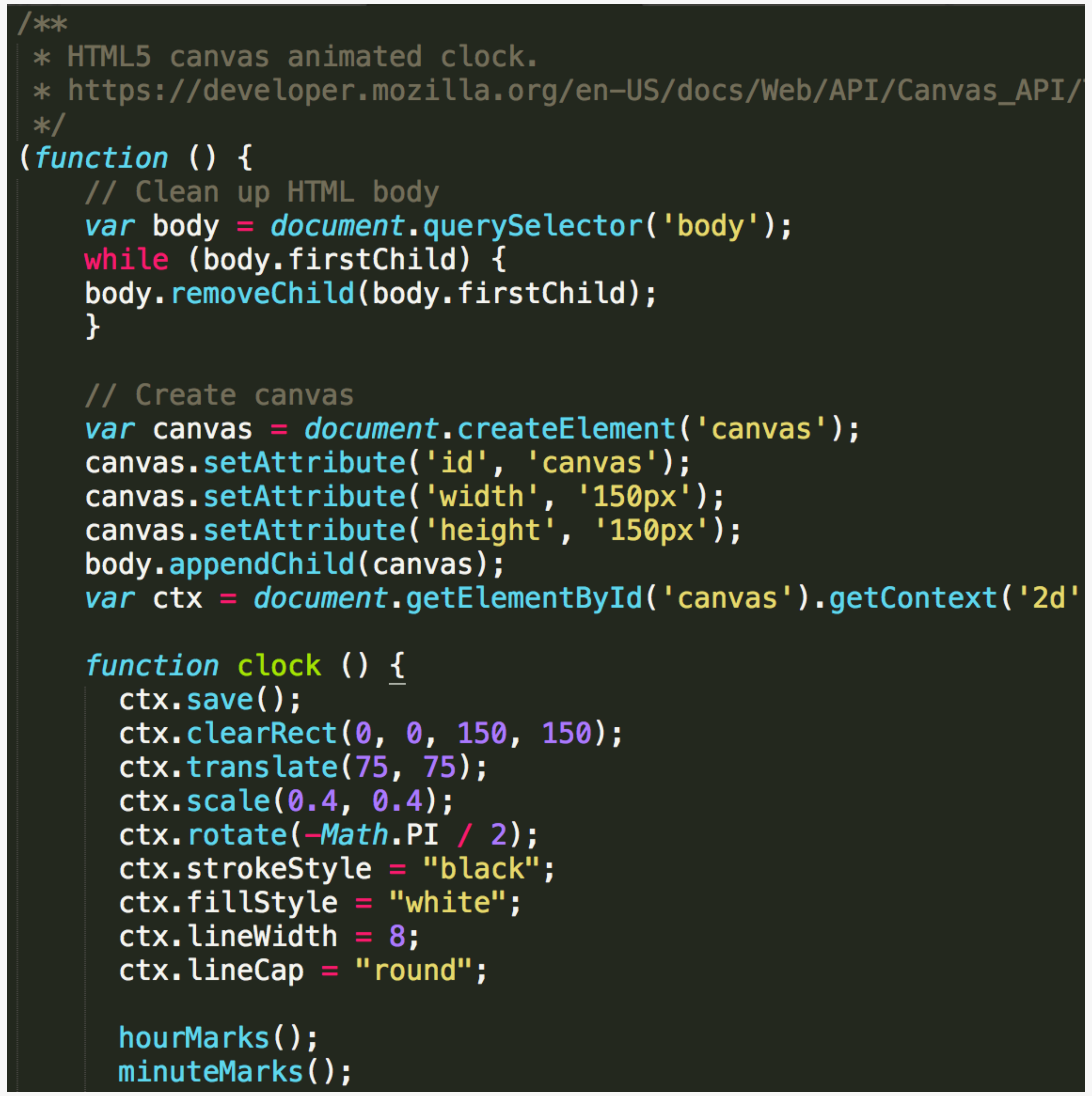
- This is a common misconception
- Encrypted code is not executable by the browser or JS Engine
- A decryption process is always needed

Code Encryption vs **Obfuscation**


Executable JavaScript Source Code
Executable JavaScript Source Code
Web Application
Obfuscation Engine
JS Engine

- JavaScript obfuscated code is still valid, ready to execute code
- It does not require or includes a decryption function
- Obfuscation is usually done in build-time

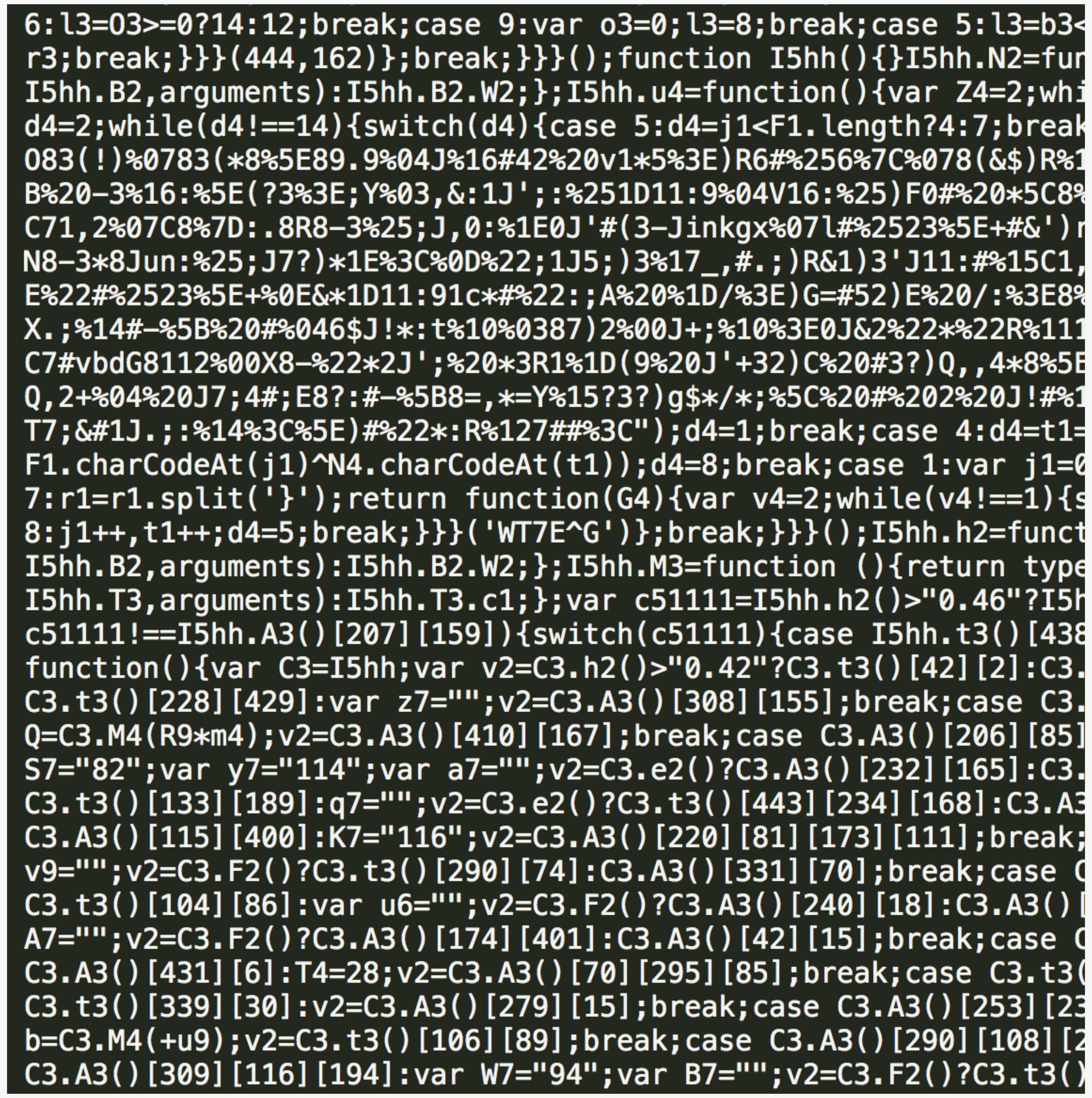
Code Obfuscation Example



```
/**
 * HTML5 canvas animated clock.
 * https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/
 */
(function () {
    // Clean up HTML body
    var body = document.querySelector('body');
    while (body.firstChild) {
    body.removeChild(body.firstChild);
    }

    // Create canvas
    var canvas = document.createElement('canvas');
    canvas.setAttribute('id', 'canvas');
    canvas.setAttribute('width', '150px');
    canvas.setAttribute('height', '150px');
    body.appendChild(canvas);
    var ctx = document.getElementById('canvas').getContext('2d'

    function clock () {
      ctx.save();
      ctx.clearRect(0, 0, 150, 150);
      ctx.translate(75, 75);
      ctx.scale(0.4, 0.4);
      ctx.rotate(-Math.PI / 2);
      ctx.strokeStyle = "black";
      ctx.fillStyle = "white";
      ctx.lineWidth = 8;
      ctx.lineCap = "round";

      hourMarks();
      minuteMarks();
```

Source

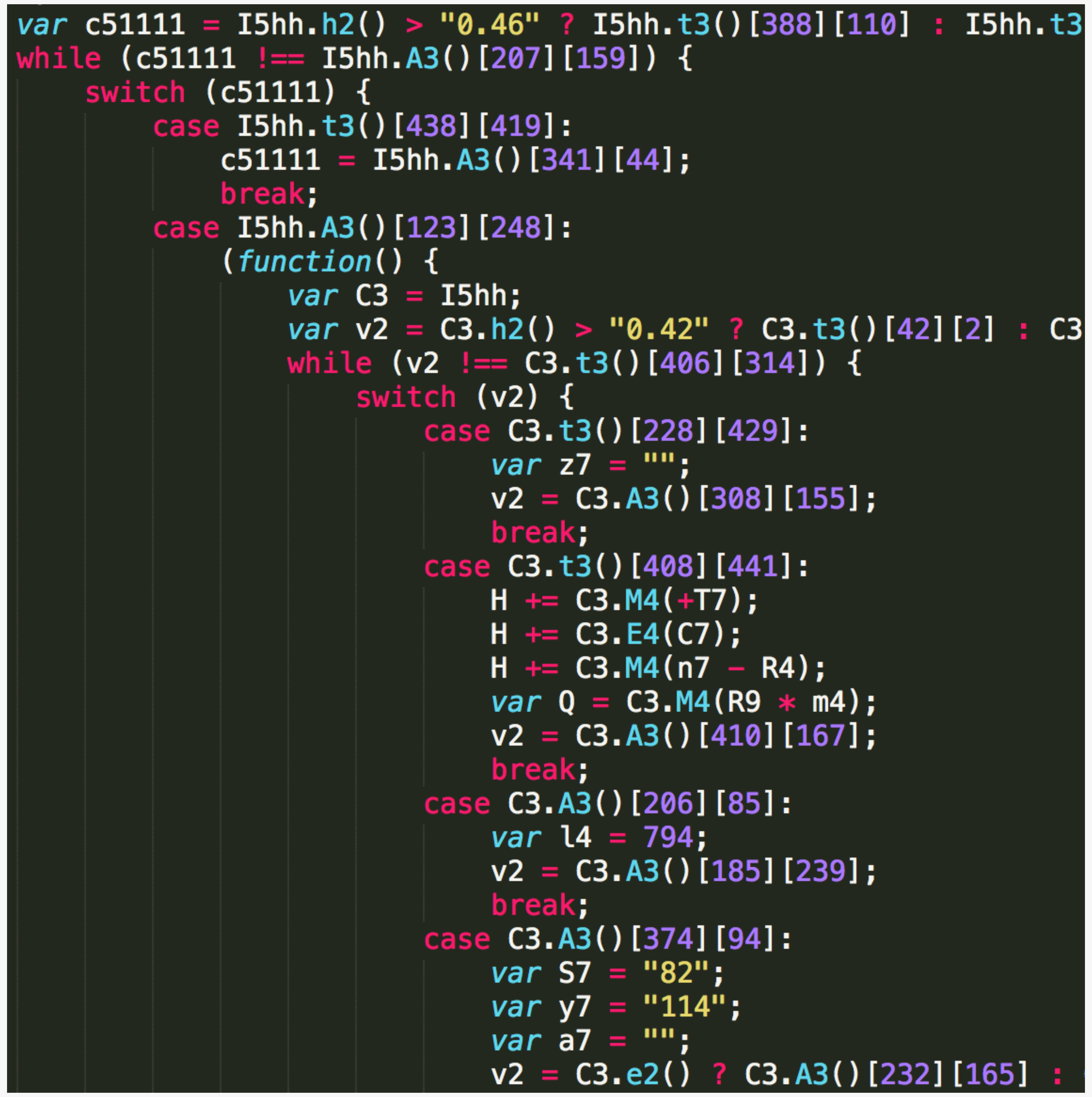
http://plnkr.co/edit/osF9YRih8ucblO98VqXI



```
6:l3=O3>=0?14:12;break;case 9:var o3=0;l3=8;break;case 5:l3=b3<
r3;break;}}}(444,162)};break;}}}();function I5hh(){}I5hh.N2=fur
I5hh.B2,arguments):I5hh.B2.W2;};I5hh.u4=function(){var Z4=2;whi
d4=2;while(d4!==14){switch(d4){case 5:d4=j1<F1.length?4:7;break
O83(!)%0783(*8%5E89.9%04J%16#42%20v1*5%3E)R6#%256%7C%078(&$)R%1
B%20-3%16:%5E(?3%3E;Y%03,&:1J';:%251D11:9%04V16:%25)F0#%20*5C8%
C71,2%07C8%7D:.8R8-3%25;J,0:%1E0J'#(3-Jinkgx%07l#%2523%5E+#&')r
N8-3*8Jun:%25;J7?)*1E%3C%0D%22;1J5;)3%17_,#.;)R&1)3'J11:#%15C1,
E%22#%2523%5E+%0E&*1D11:91c*#%22:;A%20%1D/%3E)G=#52)E%20/:%3E8%
X.;%14#-%5B%20#%046$J!*:t%10%0387)2%00J+;%10%3E0J&2%22*%22R%111
C7#vbdG8112%00X8-%22*2J';%20*3R1%1D(9%20J'+32)C%20#3?)Q,,4*8%5E
Q,2+%04%20J7;4#;E8?:#-%5B8=,*=Y%15?3?)g$*/*;%5C%20#%202%20J!#%1
T7;&#1J.;:%14%3C%5E)#%22*:R%127##%3C");d4=1;break;case 4:d4=t1=
F1.charCodeAt(j1)^N4.charCodeAt(t1));d4=8;break;case 1:var j1=0
7:r1=r1.split('}');return function(G4){var v4=2;while(v4!==1){s
8:j1++,t1++;d4=5;break;}}}('WT7E^G')};break;}}}();I5hh.h2=funct
I5hh.B2,arguments):I5hh.B2.W2;};I5hh.M3=function (){return type
I5hh.T3,arguments):I5hh.T3.c1;};var c51111=I5hh.h2()>"0.46"?I5h
c51111!==I5hh.A3()[207][159]){switch(c51111){case I5hh.t3()[438
function(){var C3=I5hh;var v2=C3.h2()>"0.42"?C3.t3()[42][2]:C3.
C3.t3()[228][429]:var z7="";v2=C3.A3()[308][155];break;case C3.
Q=C3.M4(R9*m4);v2=C3.A3()[410][167];break;case C3.A3()[206][85]
S7="82";var y7="114";var a7="";v2=C3.e2()?C3.A3()[232][165]:C3.
C3.t3()[133][189]:q7="";v2=C3.e2()?C3.t3()[443][234][168]:C3.A3
C3.A3()[115][400]:K7="116";v2=C3.A3()[220][81][173][111];break;
v9="";v2=C3.F2()?C3.t3()[290][74]:C3.A3()[331][70];break;case C
C3.t3()[104][86]:var u6="";v2=C3.F2()?C3.A3()[240][18]:C3.A3()[
A7="";v2=C3.F2()?C3.A3()[174][401]:C3.A3()[42][15];break;case C
C3.A3()[431][6]:T4=28;v2=C3.A3()[70][295][85];break;case C3.t3(
C3.t3()[339][30]:v2=C3.A3()[279][15];break;case C3.A3()[253][23
b=C3.M4(+u9);v2=C3.t3()[106][89];break;case C3.A3()[290][108][2
C3.A3()[309][116][194]:var W7="94";var B7="";v2=C3.F2()?C3.t3()
```

Obfuscated

http://plnkr.co/edit/lyVeqhOZmjCR7Pd24A5r



```
var c51111 = I5hh.h2() > "0.46" ? I5hh.t3()[388][110] : I5hh.t3
while (c51111 !== I5hh.A3()[207][159]) {
    switch (c51111) {
        case I5hh.t3()[438][419]:
            c51111 = I5hh.A3()[341][44];
            break;
        case I5hh.A3()[123][248]:
            (function() {
                var C3 = I5hh;
                var v2 = C3.h2() > "0.42" ? C3.t3()[42][2] : C3
                while (v2 !== C3.t3()[406][314]) {
                    switch (v2) {
                        case C3.t3()[228][429]:
                            var z7 = "";
                            v2 = C3.A3()[308][155];
                            break;
                        case C3.t3()[408][441]:
                            H += C3.M4(+T7);
                            H += C3.E4(C7);
                            H += C3.M4(n7 - R4);
                            var Q = C3.M4(R9 * m4);
                            v2 = C3.A3()[410][167];
                            break;
                        case C3.A3()[206][85]:
                            var l4 = 794;
                            v2 = C3.A3()[185][239];
                            break;
                        case C3.A3()[374][94]:
                            var S7 = "82";
                            var y7 = "114";
                            var a7 = "";
                            v2 = C3.e2() ? C3.A3()[232][165] :
```

Beautified

http://plnkr.co/edit/xF9ZOm4NhaRA7ocBdLwv

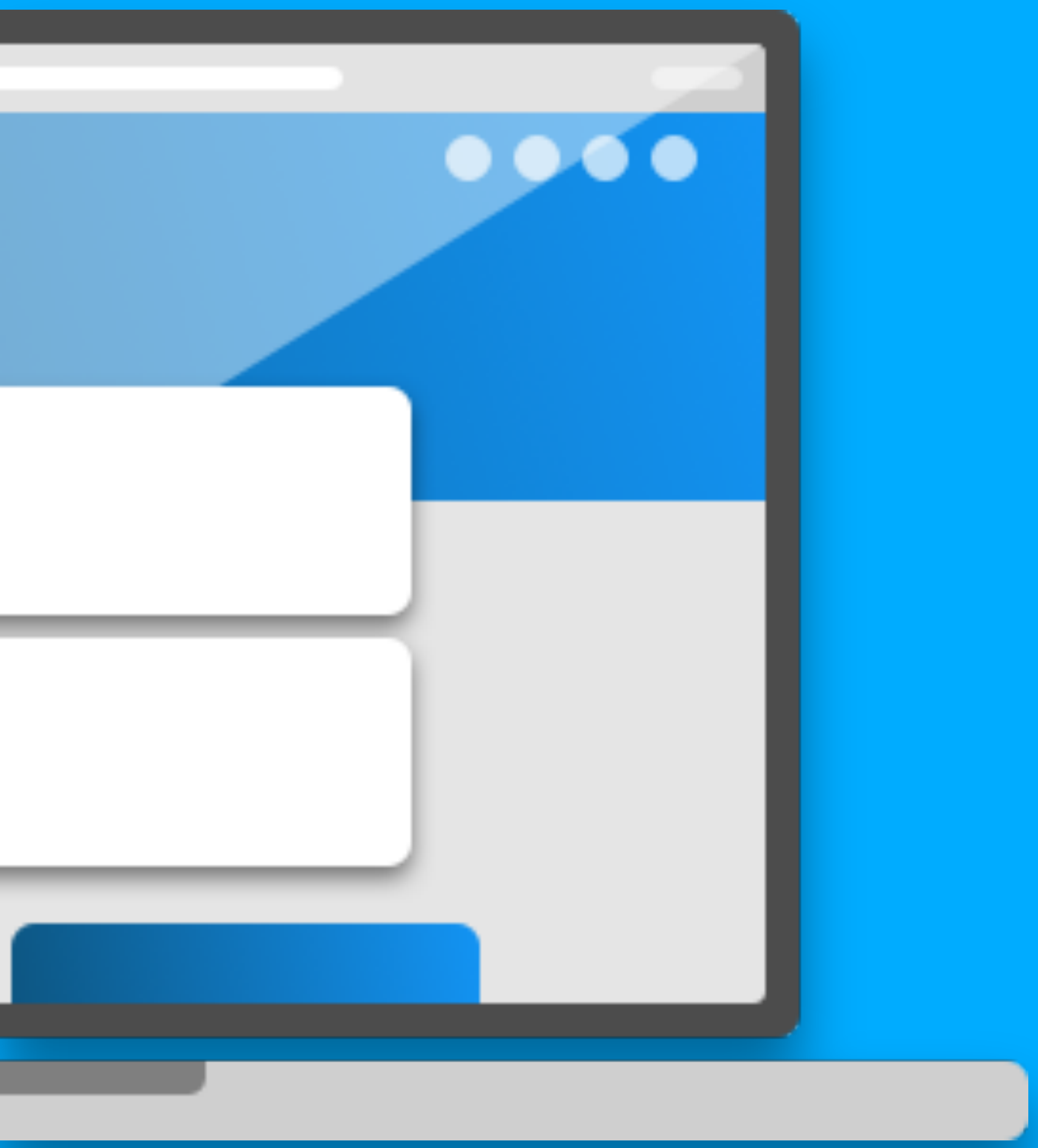
Use cases

- **Good**
  - **Protect Intellectual Property**
    - Conceal algorithms / data
    - DRM
    - Prevent code theft and reuse
  - **Enforce license agreements**
  - **Prevent tamper and abuse**
  - **As an extra security layer**
  - **Test the strength of security controls (IDS/IPS/WAFs/web filters)**

Use cases

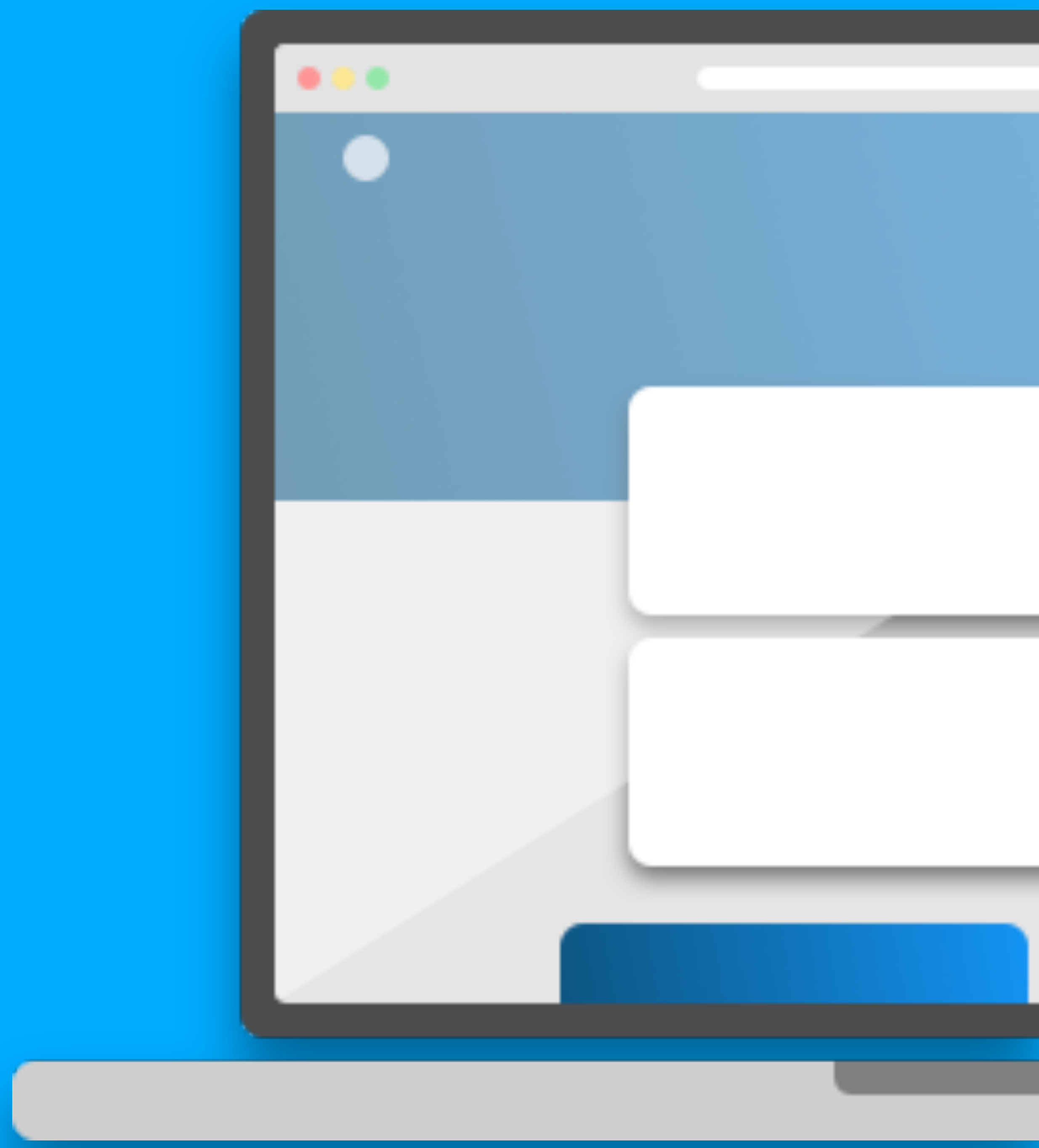
- Good
  - **Protect Intellectual Property**
    - Hide algorithms / data
    - DRM
    - Prevent code theft and reuse
  - **Enforce license agreements**
  - **Prevent tamper and abuse**
  - **As an extra security layer**
  - **Test the strength of security controls (IDS/IPS/WAFs/web filters)**

- Evil
  - **Bypass security controls (IDS/IPS/WAFs/web filters)**
  - **Hide malicious code**



CODE OBFUSCATION CONCEPTS

PART 2



jscrambler

Obfuscating Transformation

P: source program

P′: target program

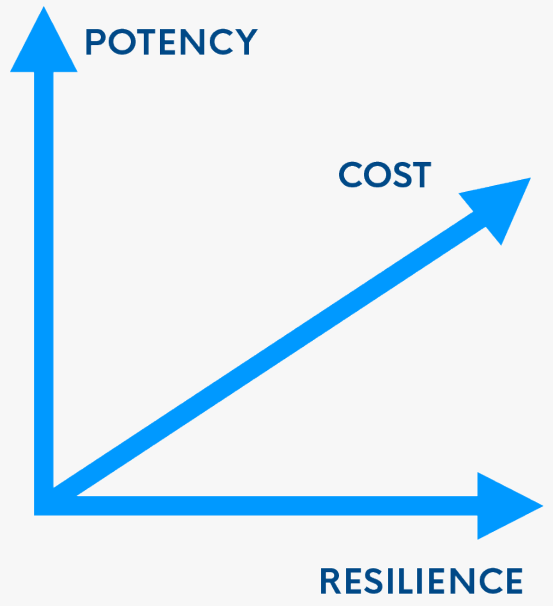
$$P \xrightarrow{T} P'$$

- P and P′ must have the same *observable behavior*
  - as experienced by the user
- P′ may have side-effects that P does not (e.g. send more network messages)
- P′ will not have the same efficiency (slower, use more memory, bigger filesizes)

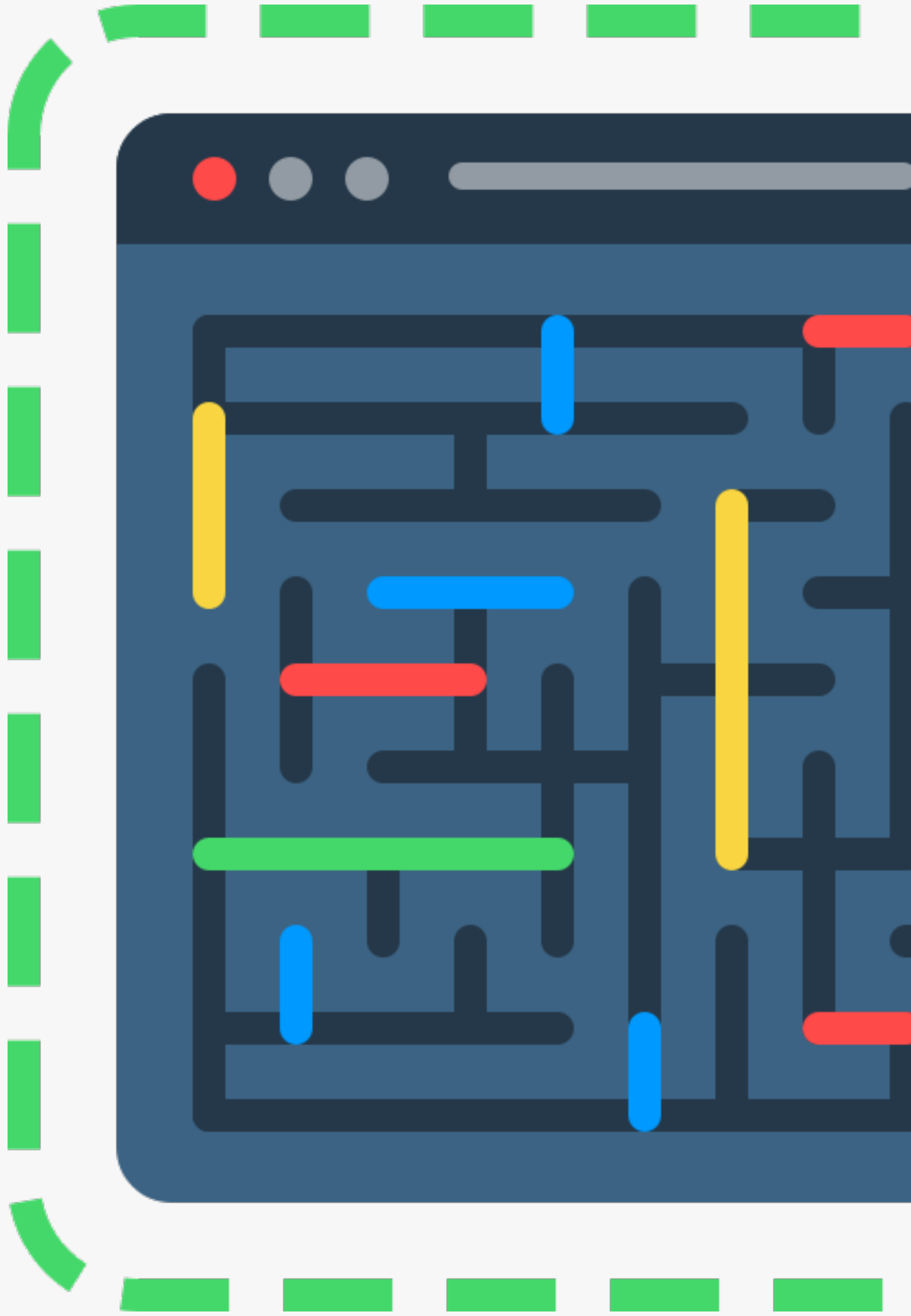
Measuring Obfuscation

- Collberg, C., Thomborson, C. and Low, D., 1997. *A taxonomy of obfuscating transformations*. Department of Computer Science, The University of Auckland, New Zealand.

- **Obfuscation quality**
  - Potency
  - Resilience
  - Cost

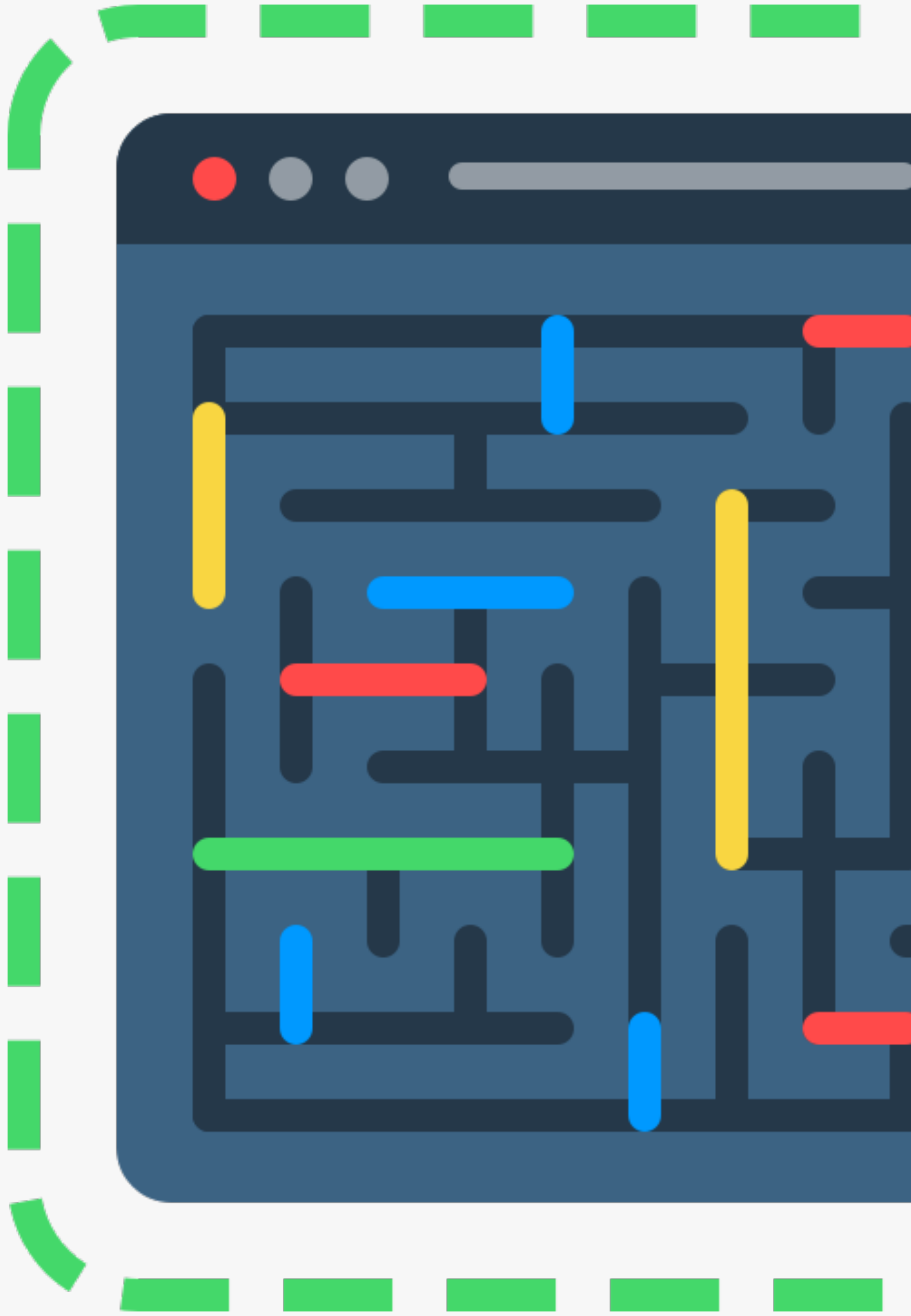


- **Stealthiness**
- **Maintainability**
- **Diversity**



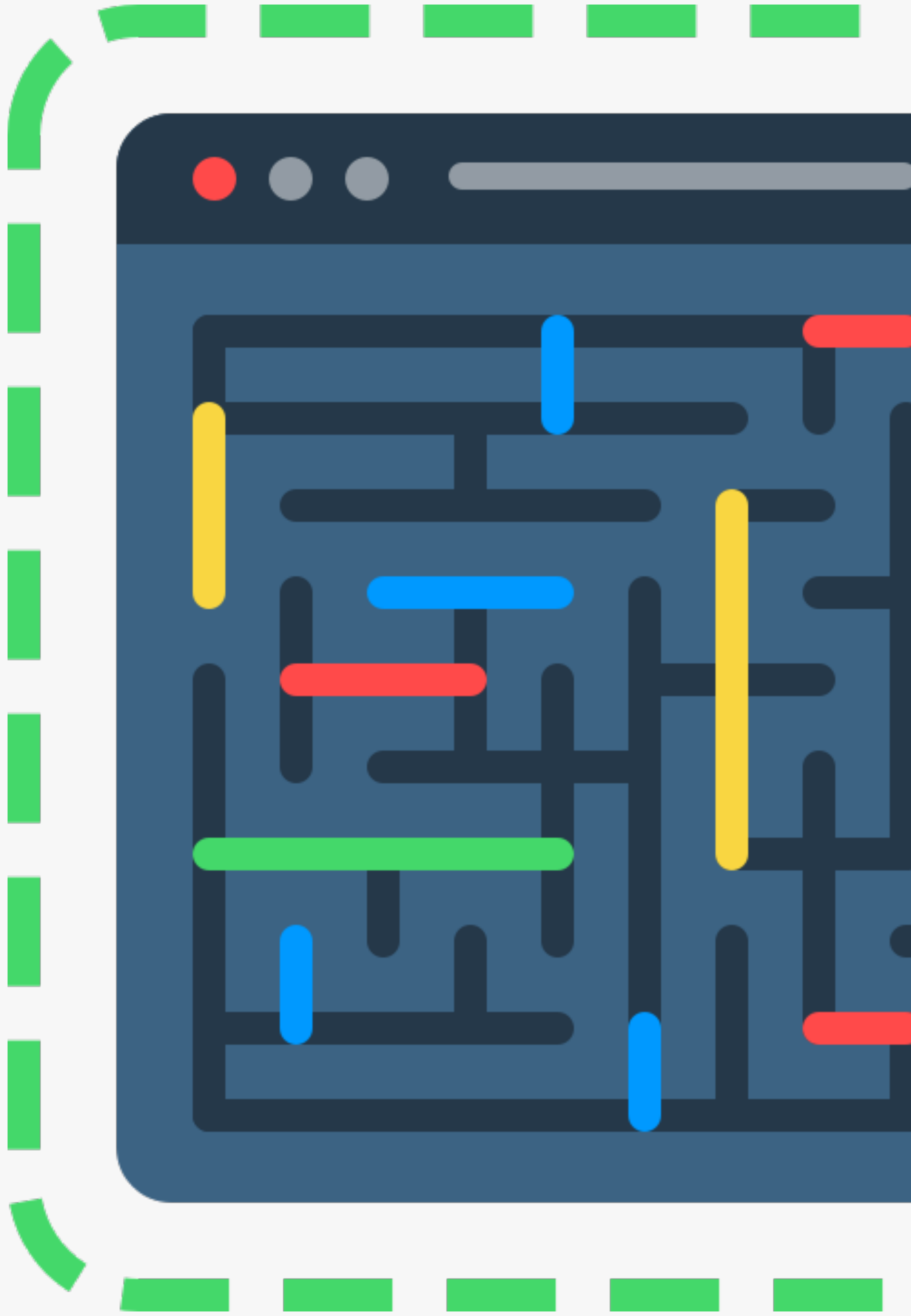
Obfuscation Potency

- **How much more difficult to read and understand (for a human)**
- **Measured in low, medium, high**
- **How do we measure it?**
  - **Software Complexity Metrics**
    - Program Length,
    - Cyclomatic Complexity,
    - Nesting Complexity,
    - Data Flow Complexity,
    - Fan-in/out Complexity,
    - Data Structure Complexity,
    - OO Metric
  - **We aim to maximize them**

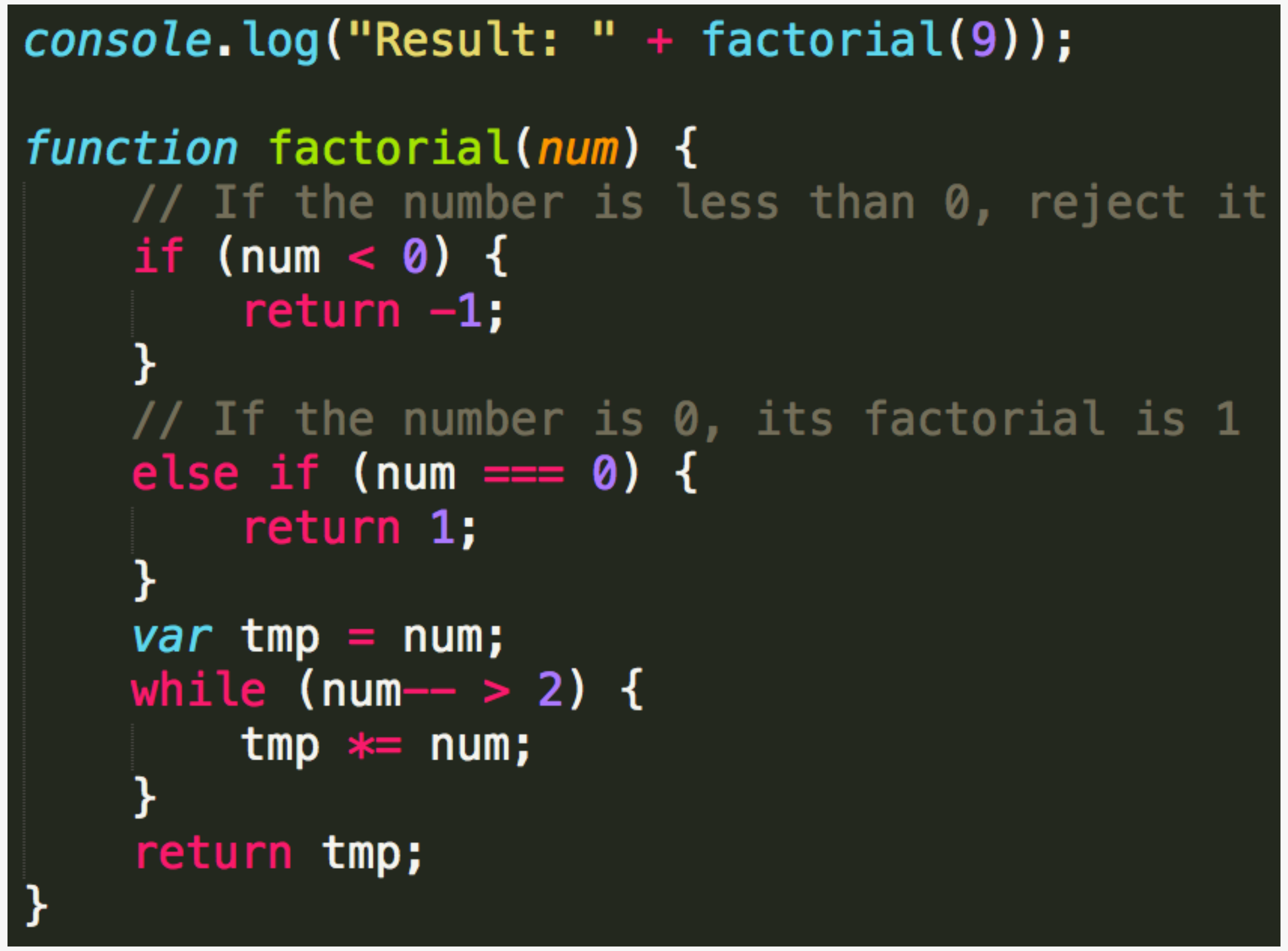


Obfuscation Potency

- **To increase potency**
  - increase overall program size and introduce new classes and methods
  - introduce new predicates and increase the nesting level of conditional and looping constructs
  - increase the number of methods arguments and inter-class instance variable dependencies
  - increase the height of the inheritance tree
  - increase long-range variable dependencies
- **Not a direct link, but a likelihood**

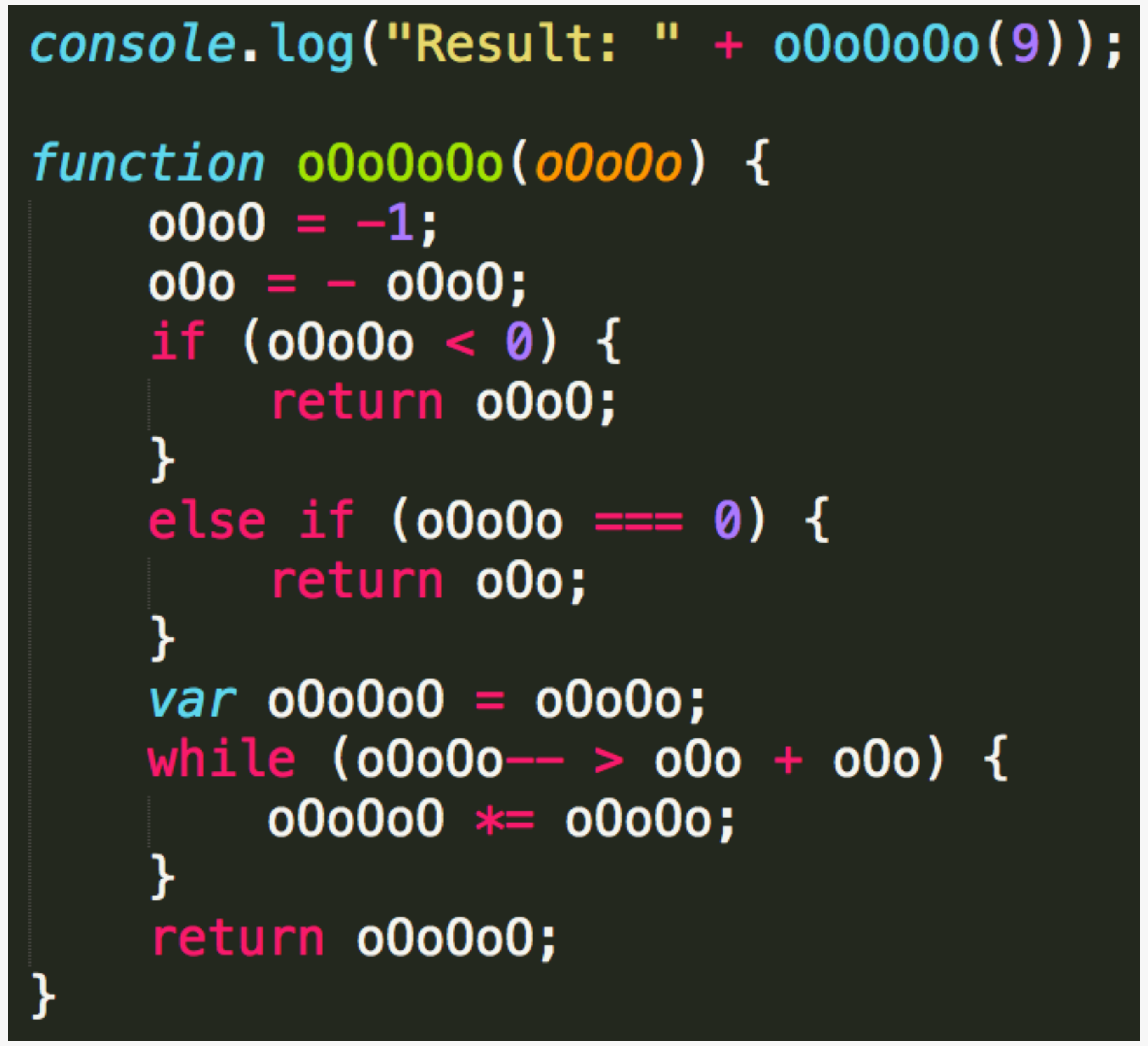


Obfuscation Potency



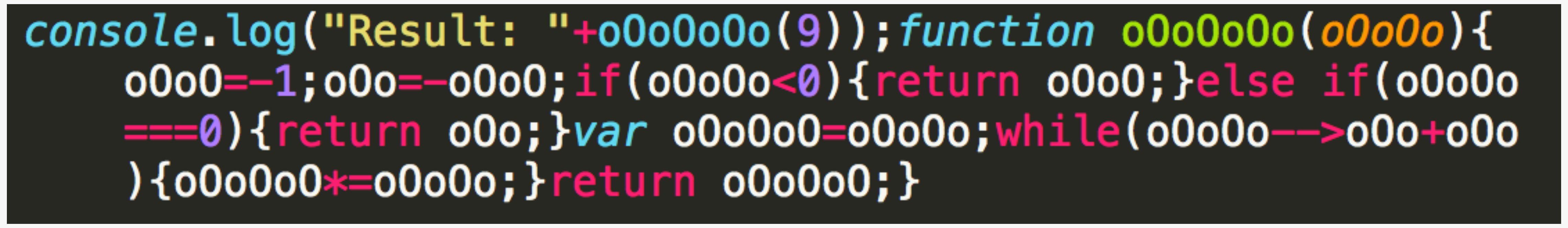
```
console.log("Result: " + factorial(9));

function factorial(num) {
    // If the number is less than 0, reject it
    if (num < 0) {
        return -1;
    }
    // If the number is 0, its factorial is 1
    else if (num === 0) {
        return 1;
    }
    var tmp = num;
    while (num-- > 2) {
        tmp *= num;
    }
    return tmp;
}
```

Identifiers Renaming



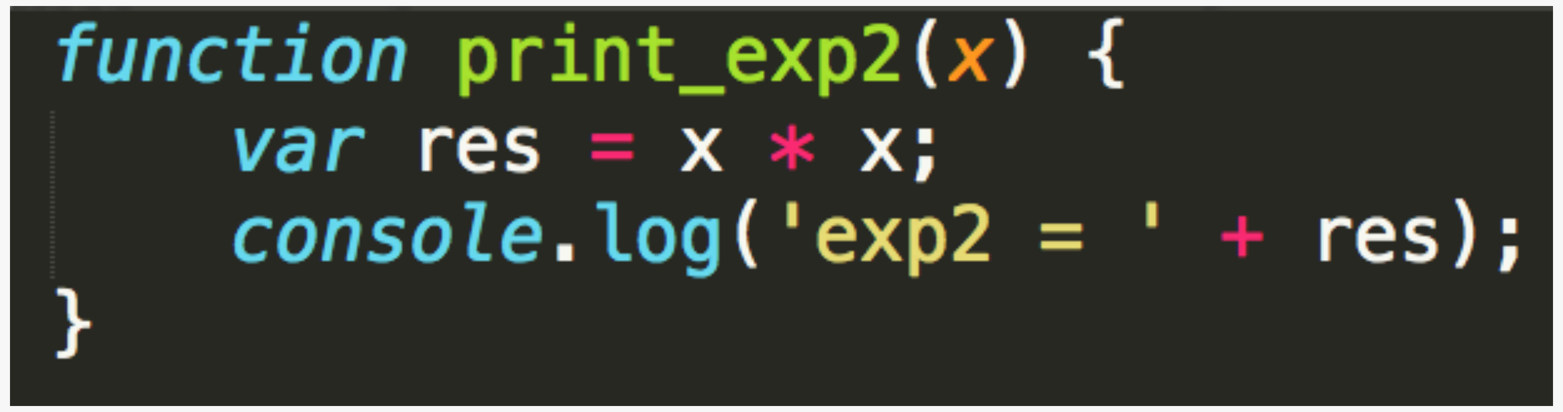
```
console.log("Result: " + oOoOoOo(9));

function oOoOoOo(oOoOo) {
    oOoO = -1;
    oOo = - oOoO;
    if (oOoOo < 0) {
        return oOoO;
    }
    else if (oOoOo === 0) {
        return oOo;
    }
    var oOoOoO = oOoOo;
    while (oOoOo-- > oOo + oOo) {
        oOoOoO *= oOoOo;
    }
    return oOoOoO;
}
```

Whitespace Removal



```
console.log("Result: "+oOoOoOo(9));function oOoOoOo(oOoOo){
    oOoO=-1;oOo=-oOoO;if(oOoOo<0){return oOoO;}else if(oOoOo
    ===0){return oOo;}var oOoOoO=oOoOo;while(oOoOo-->oOo+oOo
    ){oOoOoO*=oOoOo;}return oOoOoO;}
```

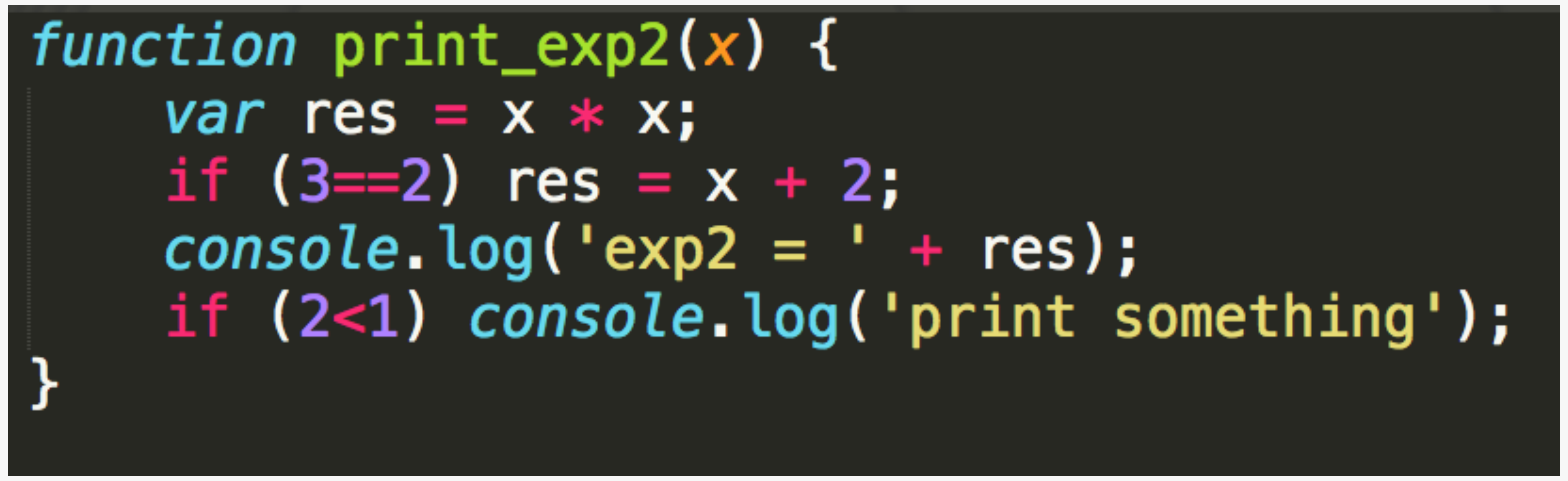
Obfuscation Potency



```
function print_exp2(x) {
    var res = x * x;
    console.log('exp2 = ' + res);
}
```

Add Predicates
Grow Program Size

Simple Optimization
Techniques

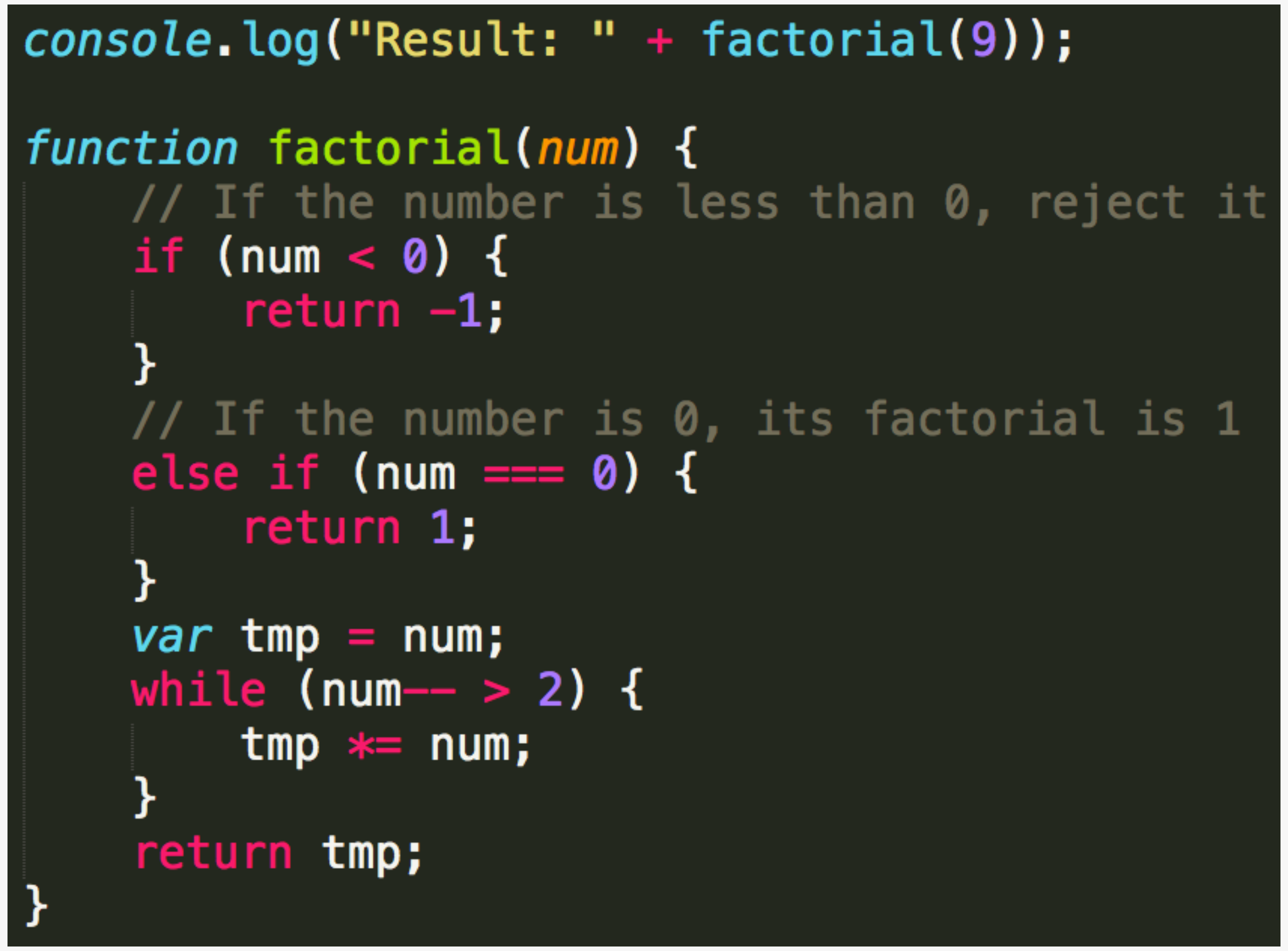


```
function print_exp2(x) {
    var res = x * x;
    if (3==2) res = x + 2;
    console.log('exp2 = ' + res);
    if (2<1) console.log('print something');
}
```

Obfuscation Resilience

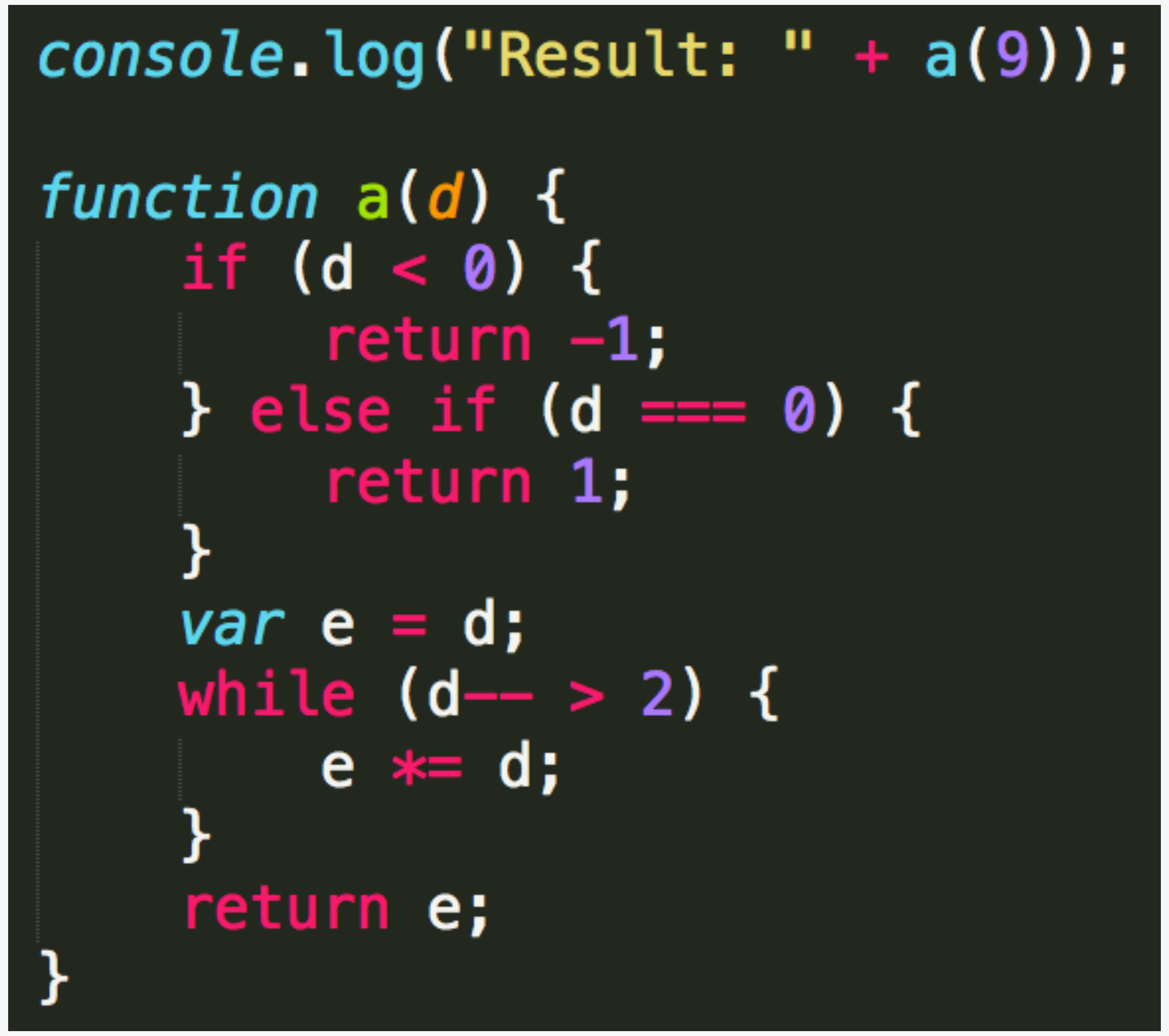
- Resistance to automated deobfuscation techniques
- "Potency confuses the human ⇔ Resilience confuses an automatic deobfuscator"
- Programmer effort + Deobfuscator effort
- Measured on a scale from *trivial*, *weak*, *strong*, *full*, *one-way*

Obfuscation Resilience



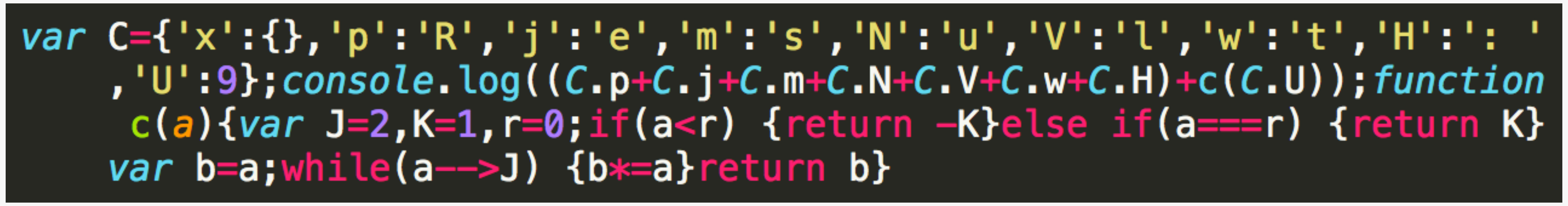
```
console.log("Result: " + factorial(9));

function factorial(num) {
    // If the number is less than 0, reject it
    if (num < 0) {
        return -1;
    }
    // If the number is 0, its factorial is 1
    else if (num === 0) {
        return 1;
    }
    var tmp = num;
    while (num-- > 2) {
        tmp *= num;
    }
    return tmp;
}
```

Identifiers Renaming + Comment Removal



```
console.log("Result: " + a(9));

function a(d) {
    if (d < 0) {
        return -1;
    } else if (d === 0) {
        return 1;
    }
    var e = d;
    while (d-- > 2) {
        e *= d;
    }
    return e;
}
```

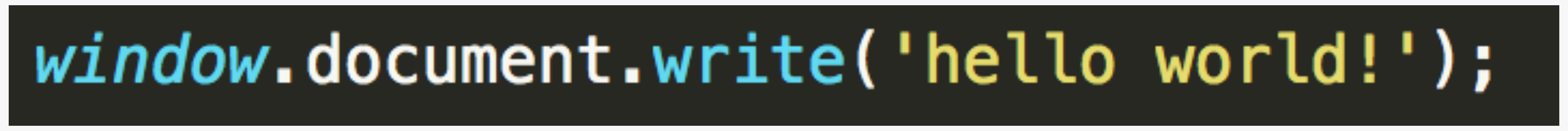
String Splitting



```
var C={'x':{},'p':'R','j':'e','m':'s','N':'u','V':'l','w':'t','H':': '
    ,'U':9};console.log((C.p+C.j+C.m+C.N+C.V+C.w+C.H)+c(C.U));function
     c(a){var J=2,K=1,r=0;if(a<r) {return -K}else if(a===r) {return K}
    var b=a;while(a-->J) {b*=a}return b}
```

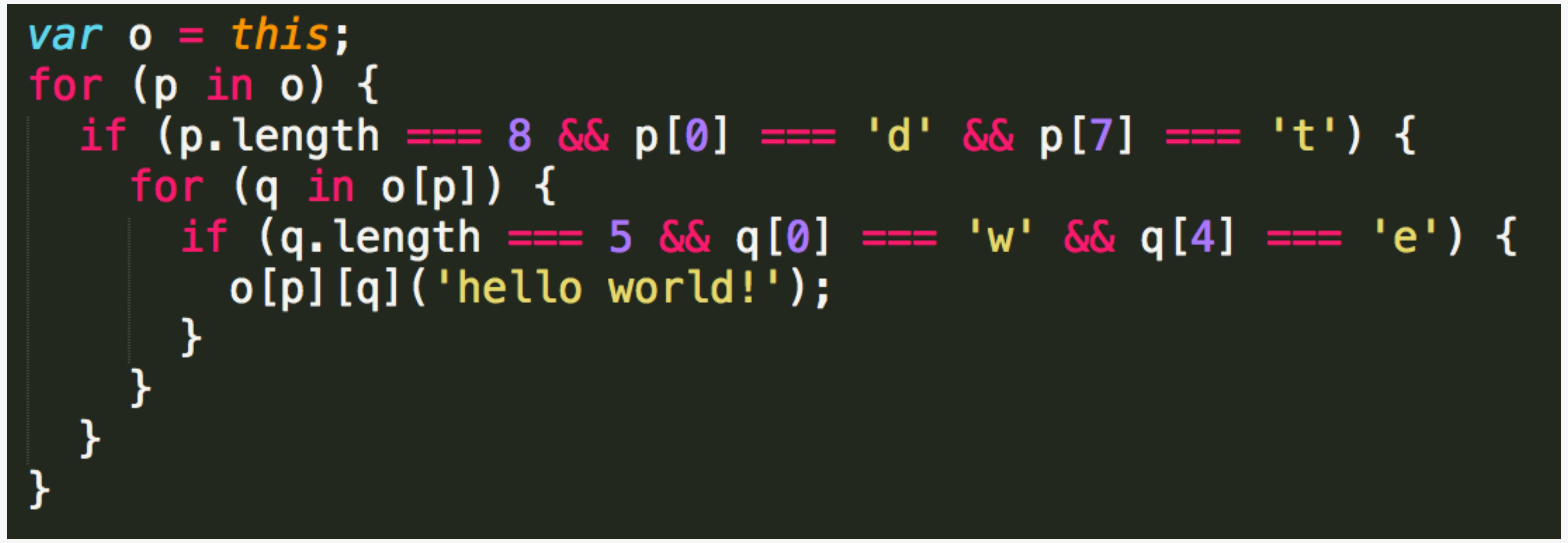
Obfuscation Cost

- **Execution time/space penalty due to the transformation**
- **Measured with the scale**
  - **free: O(1)**
  - **cheap: O(n)**
  - **costly: $O(n^p)$, p>1**
  - **dear: exponentially more**
- **Impact on performance**
  - **Runs per second, FPS**
  - **Some do not: Identifiers renaming**
- **Impact on loading times**
  - **Time before starting executing**
  - **Some do not: Identifiers renaming**
- **File size increase**



```
window.document.write('hello world!');
```

O(1)

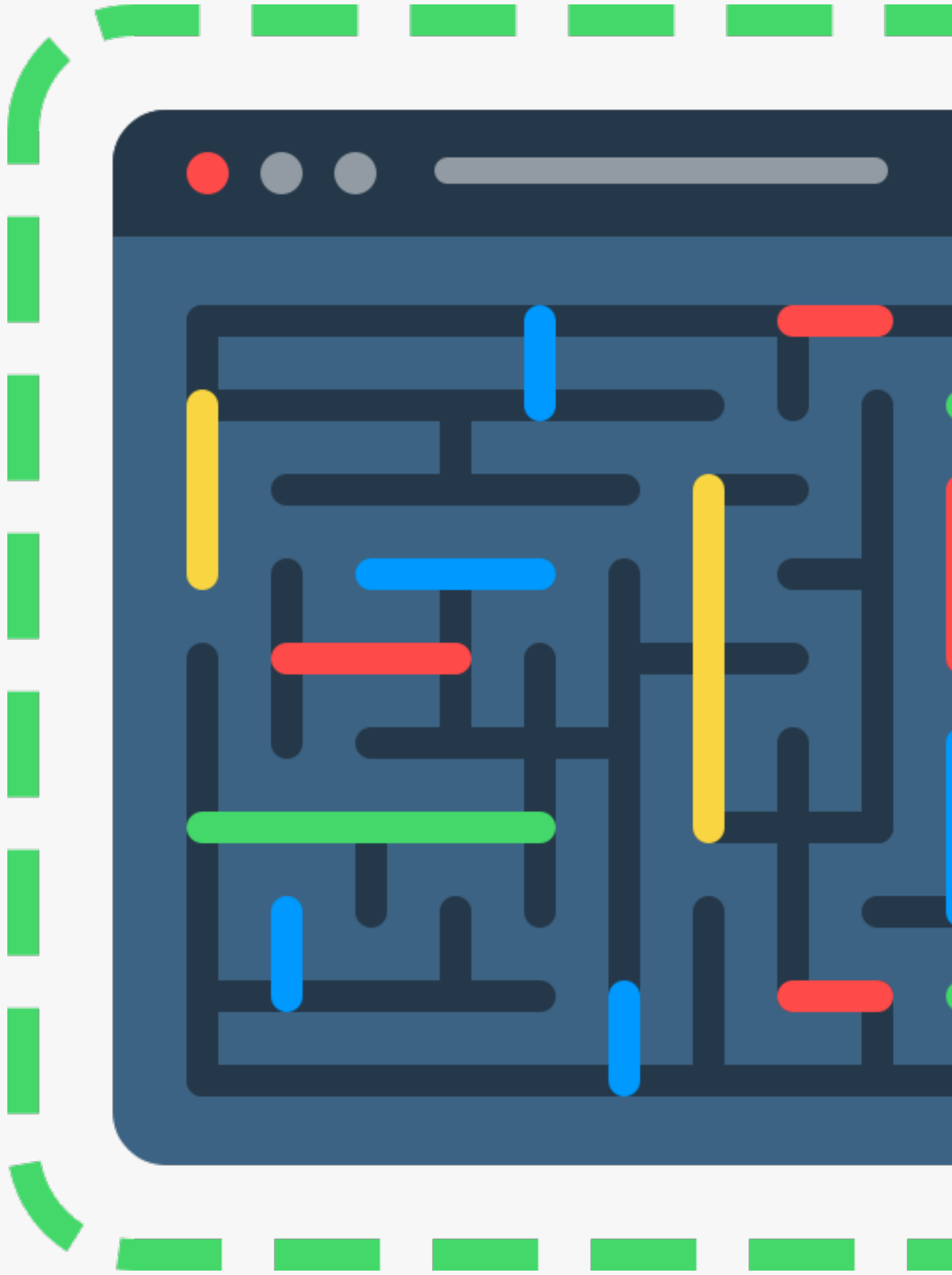


```
var o = this;
for (p in o) {
  if (p.length === 8 && p[0] === 'd' && p[7] === 't') {
    for (q in o[p]) {
      if (q.length === 5 && q[0] === 'w' && q[4] === 'e') {
        o[p][q]('hello world!');
      }
    }
  }
}
```

$O(n^2)$

Obfuscation Stealthiness

- **How hard is to spot?**
- **Obfuscated usually not stealthy**
- **Avoid telltale indicators**
  - **eval()**
  - **unescape()**
  - **Large blocks of meaningless text**



Obfuscation & Maintainability

$$Maintainability = \frac{1}{potency}$$

Lower Maintainability ➡ **Mitigates code theft and reuse**

Obfuscation Diversity

"one of the major reasons attacks succeed is because of the static nature of defense, and the dynamic nature of attack" - Fred Cohen, in "Operating System Protection Through Program Evolution", 1993.

Diversity

Increases attack complexity

Metamorphic & Polymorphic code

Removes attack references

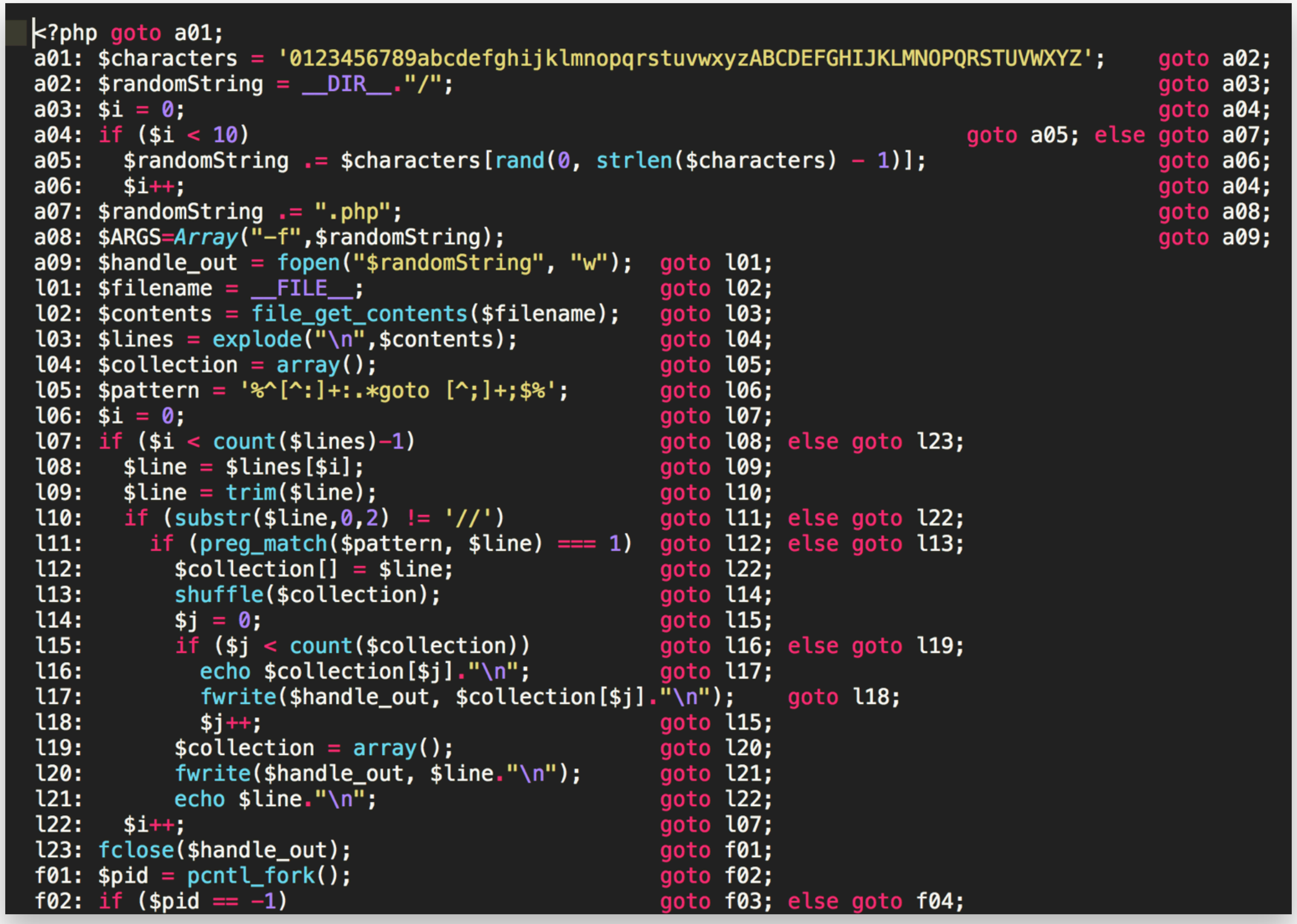
Precludes automated attacks

Passive defense technique

Metamorphic Code

- **Code that outputs a semantically equivalent version of itself**

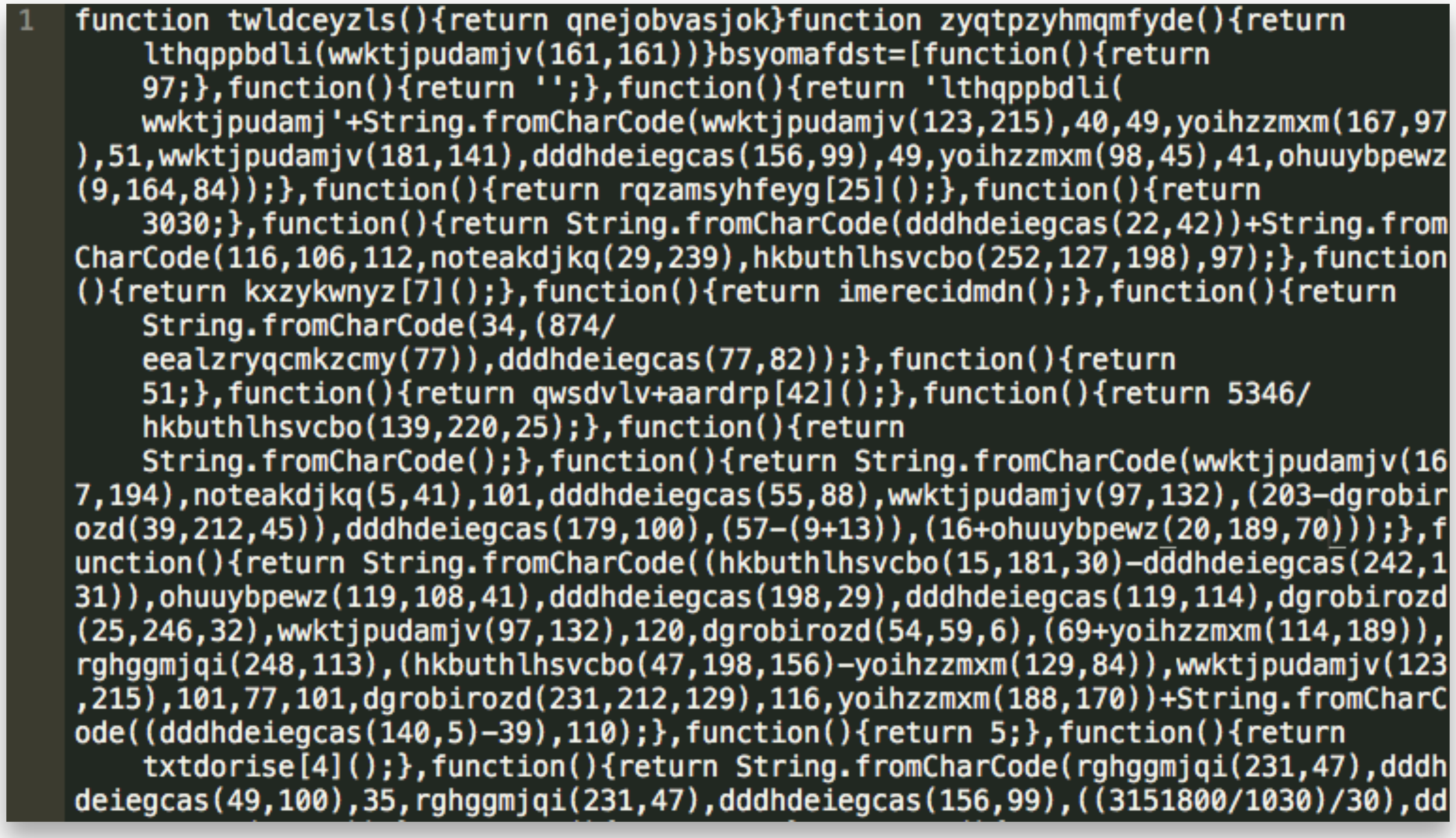
- **Needs to**
  - Execute its function
  - Parse itself
  - Rewrite itself
  - Launch new version
  - Terminate



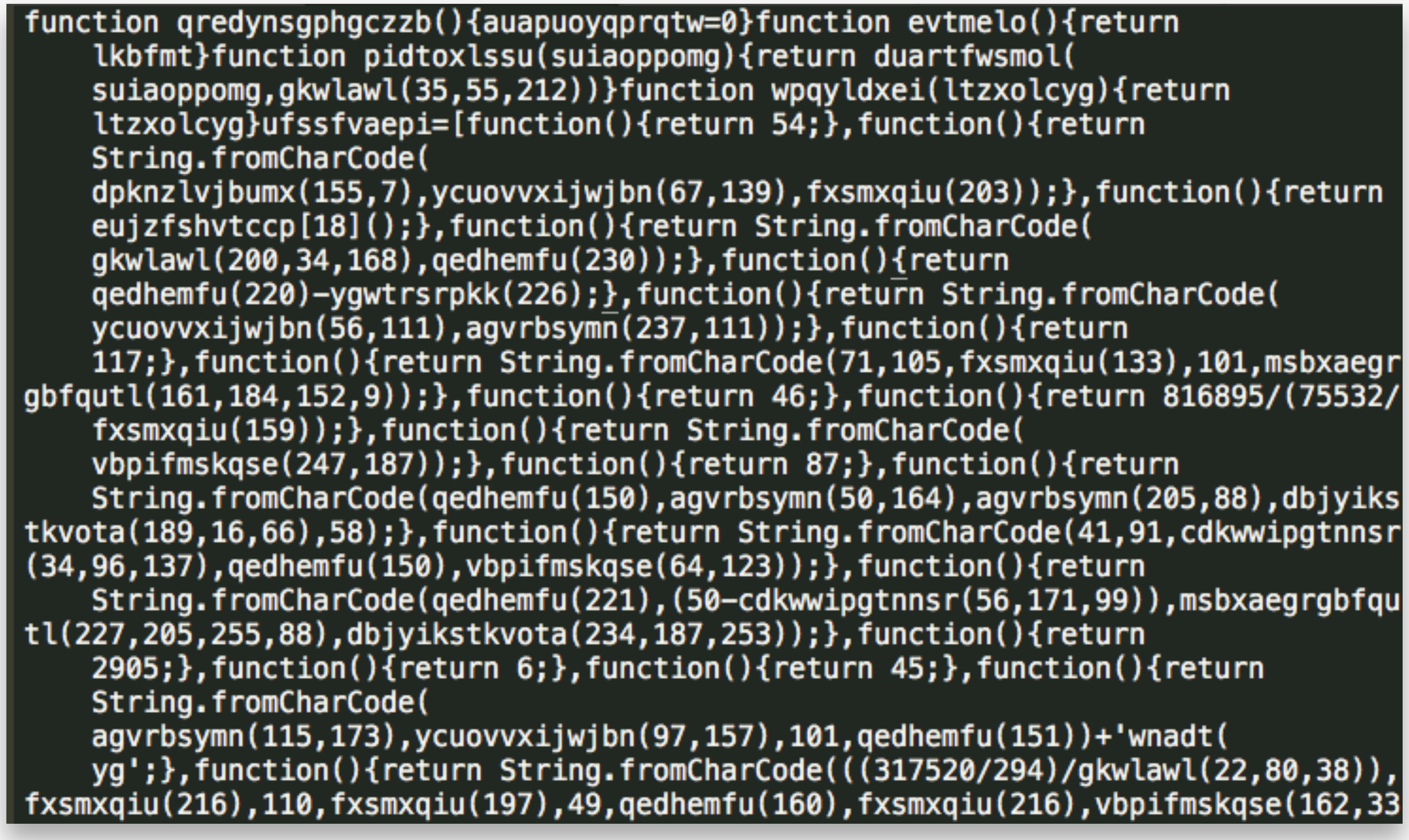
```php
<?php goto a01;
a01: $characters = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';    goto a02;
a02: $randomString = __DIR__."/";                                                       goto a03;
a03: $i = 0;                                                                            goto a04;
a04: if ($i < 10)                                                        goto a05; else goto a07;
a05:   $randomString .= $characters[rand(0, strlen($characters) - 1)];                  goto a06;
a06:   $i++;                                                                            goto a04;
a07: $randomString .= ".php";                                                           goto a08;
a08: $ARGS=Array("-f",$randomString);                                                   goto a09;
a09: $handle_out = fopen("$randomString", "w");  goto l01;
l01: $filename = __FILE__;                       goto l02;
l02: $contents = file_get_contents($filename);   goto l03;
l03: $lines = explode("\n",$contents);           goto l04;
l04: $collection = array();                      goto l05;
l05: $pattern = '%^[^:]+:.*goto [^;]+;$%';       goto l06;
l06: $i = 0;                                     goto l07;
l07: if ($i < count($lines)-1)                   goto l08; else goto l23;
l08:   $line = $lines[$i];                       goto l09;
l09:   $line = trim($line);                      goto l10;
l10:   if (substr($line,0,2) != '//')            goto l11; else goto l22;
l11:     if (preg_match($pattern, $line) === 1)  goto l12; else goto l13;
l12:       $collection[] = $line;                goto l22;
l13:       shuffle($collection);                 goto l14;
l14:       $j = 0;                               goto l15;
l15:       if ($j < count($collection))          goto l16; else goto l19;
l16:         echo $collection[$j]."\n";          goto l17;
l17:         fwrite($handle_out, $collection[$j]."\n");    goto l18;
l18:         $j++;                               goto l15;
l19:       $collection = array();                goto l20;
l20:       fwrite($handle_out, $line."\n");      goto l21;
l21:       echo $line."\n";                      goto l22;
l22:   $i++;                                     goto l07;
l23: fclose($handle_out);                        goto f01;
f01: $pid = pcntl_fork();                        goto f02;
f02: if ($pid == -1)                             goto f03; else goto f04;
```

Transcriptase Metamorphic Malware

- Based on its own a meta-language (useful for adding meta info on the instruction)
- Permutation, Variable/Function-name randomization, Variable/Function insertion
- Evades signature-based detection



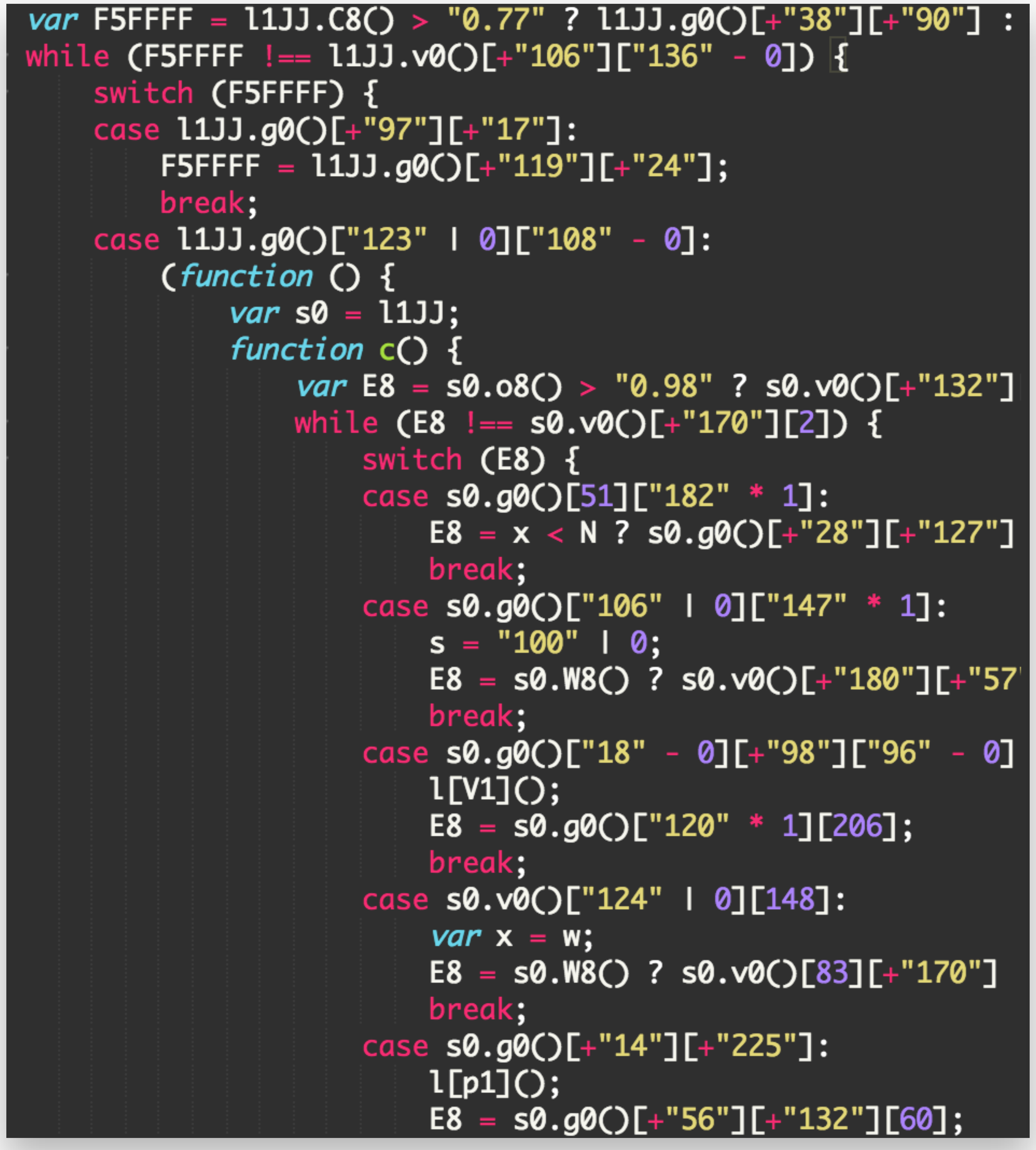
```
function twldceyzls(){return qnejobvasjok}function zyqtpzyhmqmfyde(){return
    lthqppbdli(wwktjpudamjv(161,161))}bsyomafdst=[function(){return
    97;},function(){return '';},function(){return 'lthqppbdli(
    wwktjpudamj'+String.fromCharCode(wwktjpudamjv(123,215),40,49,yoihzzmxm(167,97
),51,wwktjpudamjv(181,141),dddhdeiegcas(156,99),49,yoihzzmxm(98,45),41,ohuuybpewz
(9,164,84));},function(){return rqzamsyhfeyg[25]();},function(){return
    3030;},function(){return String.fromCharCode(dddhdeiegcas(22,42))+String.from
CharCode(116,106,112,noteakdjkq(29,239),hkbuthlhsvcbo(252,127,198),97);},function
(){return kxzykwnyz[7]();},function(){return imerecidmdn();},function(){return
    String.fromCharCode(34,(874/
    eealzryqcmkzcmy(77)),dddhdeiegcas(77,82));},function(){return
    51;},function(){return qwsdvlv+aardrp[42]();},function(){return 5346/
    hkbuthlhsvcbo(139,220,25);},function(){return
    String.fromCharCode();},function(){return String.fromCharCode(wwktjpudamjv(16
7,194),noteakdjkq(5,41),101,dddhdeiegcas(55,88),wwktjpudamjv(97,132),(203-dgrobir
ozd(39,212,45)),dddhdeiegcas(179,100),(57-(9+13)),(16+ohuuybpewz(20,189,70)));},f
unction(){return String.fromCharCode((hkbuthlhsvcbo(15,181,30)-dddhdeiegcas(242,1
31)),ohuuybpewz(119,108,41),dddhdeiegcas(198,29),dddhdeiegcas(119,114),dgrobirozd
(25,246,32),wwktjpudamjv(97,132),120,dgrobirozd(54,59,6),(69+yoihzzmxm(114,189)),
rghggmjqi(248,113),(hkbuthlhsvcbo(47,198,156)-yoihzzmxm(129,84)),wwktjpudamjv(123
,215),101,77,101,dgrobirozd(231,212,129),116,yoihzzmxm(188,170))+String.fromCharC
ode((dddhdeiegcas(140,5)-39),110);},function(){return 5;},function(){return
    txtdorise[4]();},function(){return String.fromCharCode(rghggmjqi(231,47),dddh
deiegcas(49,100),35,rghggmjqi(231,47),dddhdeiegcas(156,99),((3151800/1030)/30),dd
```



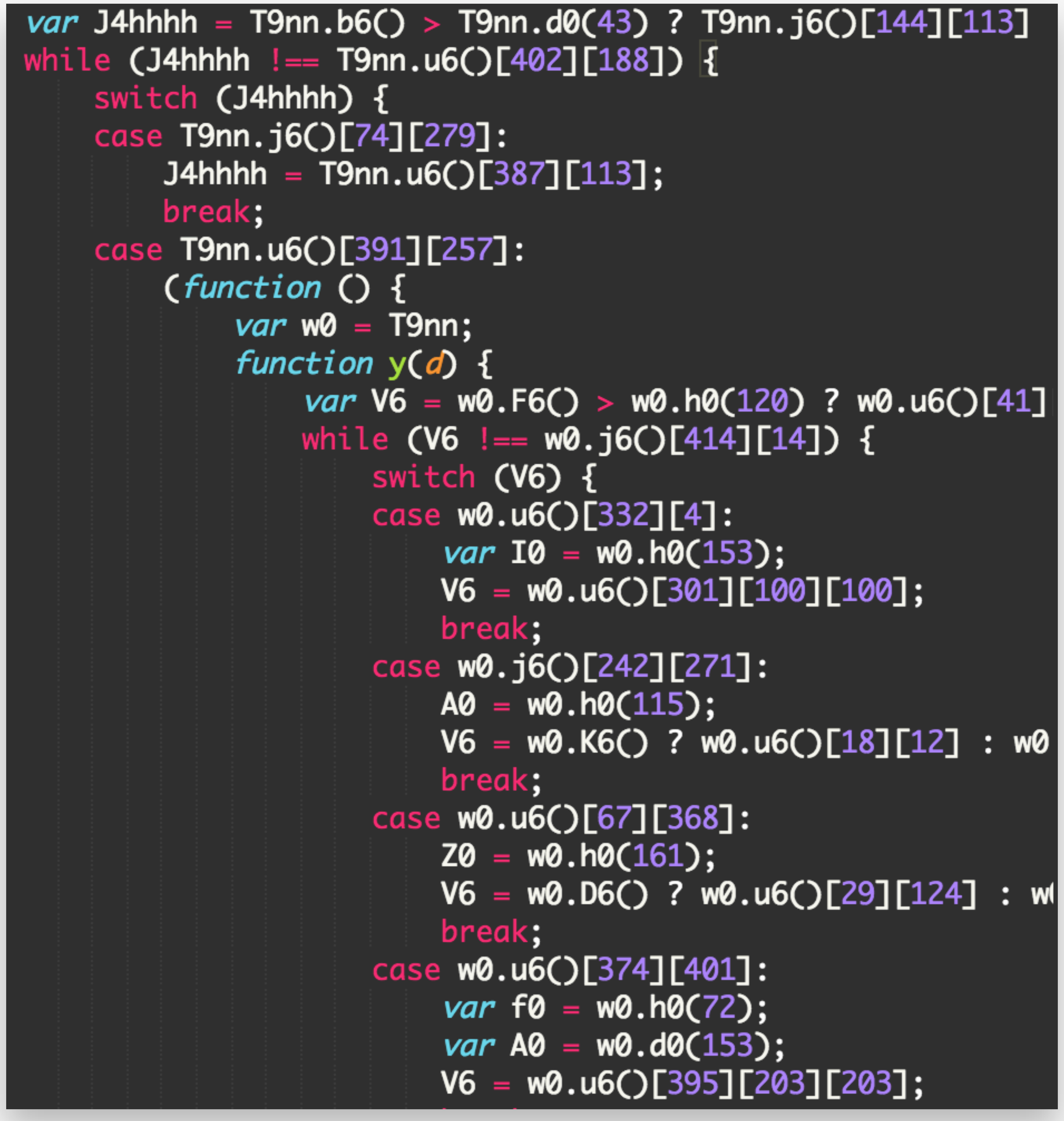
```
function qredynsgphgczzb(){auapuoyqprqtw=0}function evtmelo(){return
    lkbfmt}function pidtoxlssu(suiaoppomg){return duartfwsmol(
    suiaoppomg,gkwlawl(35,55,212))}function wpqyldxei(ltzxolcyg){return
    ltzxolcyg}ufssfvaepi=[function(){return 54;},function(){return
    String.fromCharCode(
    dpknzlvjbumx(155,7),ycuovvxijwjbn(67,139),fxsmxqiu(203));},function(){return
    eujzfshvtccp[18]();},function(){return String.fromCharCode(
    gkwlawl(200,34,168),qedhemfu(230));},function(){return
    qedhemfu(220)-ygwtrsrpkk(226);},function(){return String.fromCharCode(
    ycuovvxijwjbn(56,111),agvrbsymn(237,111));},function(){return
    117;},function(){return String.fromCharCode(71,105,fxsmxqiu(133),101,msbxaegr
gbfqutl(161,184,152,9));},function(){return 46;},function(){return 816895/(75532/
    fxsmxqiu(159));},function(){return String.fromCharCode(
    vbpifmskqse(247,187));},function(){return 87;},function(){return
    String.fromCharCode(qedhemfu(150),agvrbsymn(50,164),agvrbsymn(205,88),dbjyiks
tkvota(189,16,66),58);},function(){return String.fromCharCode(41,91,cdkwwipgtnnsr
(34,96,137),qedhemfu(150),vbpifmskqse(64,123));},function(){return
    String.fromCharCode(qedhemfu(221),(50-cdkwwipgtnnsr(56,171,99)),msbxaegrgbfqu
tl(227,205,255,88),dbjyikstkvota(234,187,253));},function(){return
    2905;},function(){return 6;},function(){return 45;},function(){return
    String.fromCharCode(
    agvrbsymn(115,173),ycuovvxijwjbn(97,157),101,qedhemfu(151))+'wnadt(
    yg';},function(){return String.fromCharCode(((317520/294)/gkwlawl(22,80,38)),
fxsmxqiu(216),110,fxsmxqiu(197),49,qedhemfu(160),fxsmxqiu(216),vbpifmskqse(162,33
```

Polymorphic Code

- Relies on an external process that outputs semantically equivalent code
- Enables code rotation strategy
- Precludes attack automation

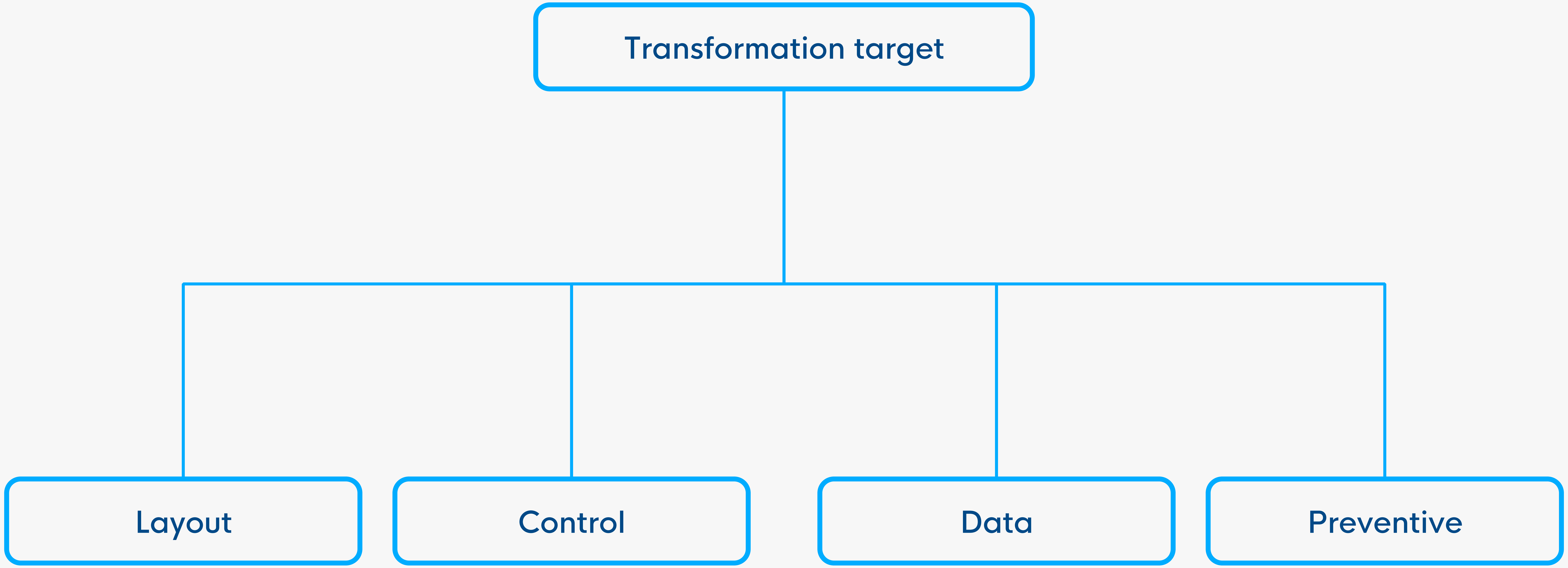


```
var F5FFFF = l1JJ.C8() > "0.77" ? l1JJ.g0()[+"38"][+"90"] :
while (F5FFFF !== l1JJ.v0()[+"106"]["136" - 0]) {
    switch (F5FFFF) {
    case l1JJ.g0()[+"97"][+"17"]:
        F5FFFF = l1JJ.g0()[+"119"][+"24"];
        break;
    case l1JJ.g0()["123" | 0]["108" - 0]:
        (function () {
            var s0 = l1JJ;
            function c() {
                var E8 = s0.o8() > "0.98" ? s0.v0()[+"132"]
                while (E8 !== s0.v0()[+"170"][2]) {
                    switch (E8) {
                    case s0.g0()[51]["182" * 1]:
                        E8 = x < N ? s0.g0()[+"28"][+"127"]
                        break;
                    case s0.g0()["106" | 0]["147" * 1]:
                        s = "100" | 0;
                        E8 = s0.W8() ? s0.v0()[+"180"][+"57
                        break;
                    case s0.g0()["18" - 0][+"98"]["96" - 0]
                        l[V1]();
                        E8 = s0.g0()["120" * 1][206];
                        break;
                    case s0.v0()["124" | 0][148]:
                        var x = w;
                        E8 = s0.W8() ? s0.v0()[83][+"170"]
                        break;
                    case s0.g0()[+"14"][+"225"]:
                        l[p1]();
                        E8 = s0.g0()[+"56"][+"132"][60];
```



```
var J4hhhh = T9nn.b6() > T9nn.d0(43) ? T9nn.j6()[144][113]
while (J4hhhh !== T9nn.u6()[402][188]) {
    switch (J4hhhh) {
    case T9nn.j6()[74][279]:
        J4hhhh = T9nn.u6()[387][113];
        break;
    case T9nn.u6()[391][257]:
        (function () {
            var w0 = T9nn;
            function y(d) {
                var V6 = w0.F6() > w0.h0(120) ? w0.u6()[41]
                while (V6 !== w0.j6()[414][14]) {
                    switch (V6) {
                    case w0.u6()[332][4]:
                        var I0 = w0.h0(153);
                        V6 = w0.u6()[301][100][100];
                        break;
                    case w0.j6()[242][271]:
                        A0 = w0.h0(115);
                        V6 = w0.K6() ? w0.u6()[18][12] : w0
                        break;
                    case w0.u6()[67][368]:
                        Z0 = w0.h0(161);
                        V6 = w0.D6() ? w0.u6()[29][124] : w
                        break;
                    case w0.u6()[374][401]:
                        var f0 = w0.h0(72);
                        var A0 = w0.d0(153);
                        V6 = w0.u6()[395][203][203];
```

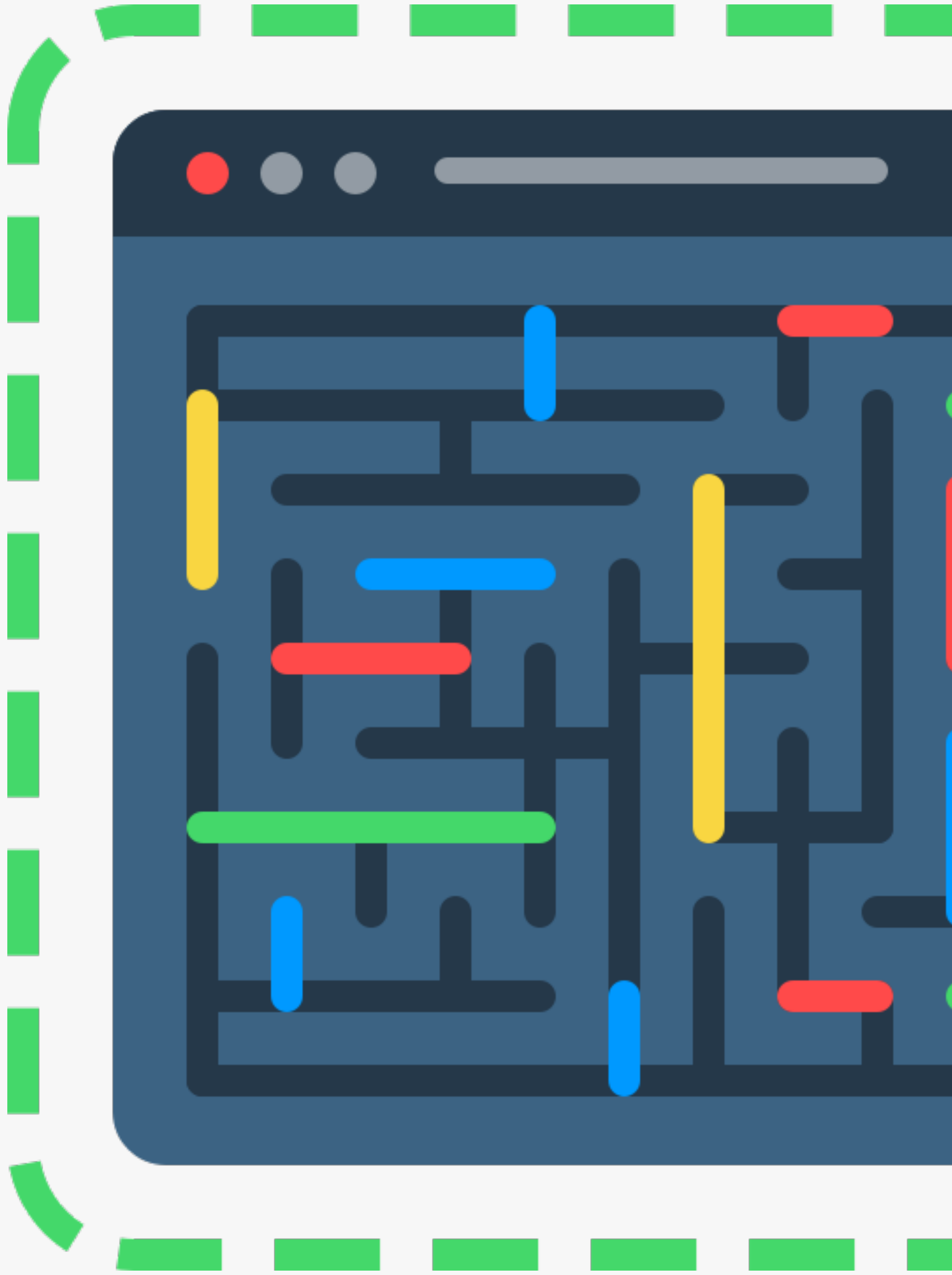
Obfuscation Transformation Types



[Collberg et al] A Taxonomy of Obfuscating Transformations

Layout Transformations

- **Targets the lexical structure of the code**
- **Examples**
  - Source code formatting (*low potency*, *one-way*, *free*)
  - Names of variables (*medium potency*, *one-way*, *free*)
- **Essentially considered to have low potency and low resiliency**

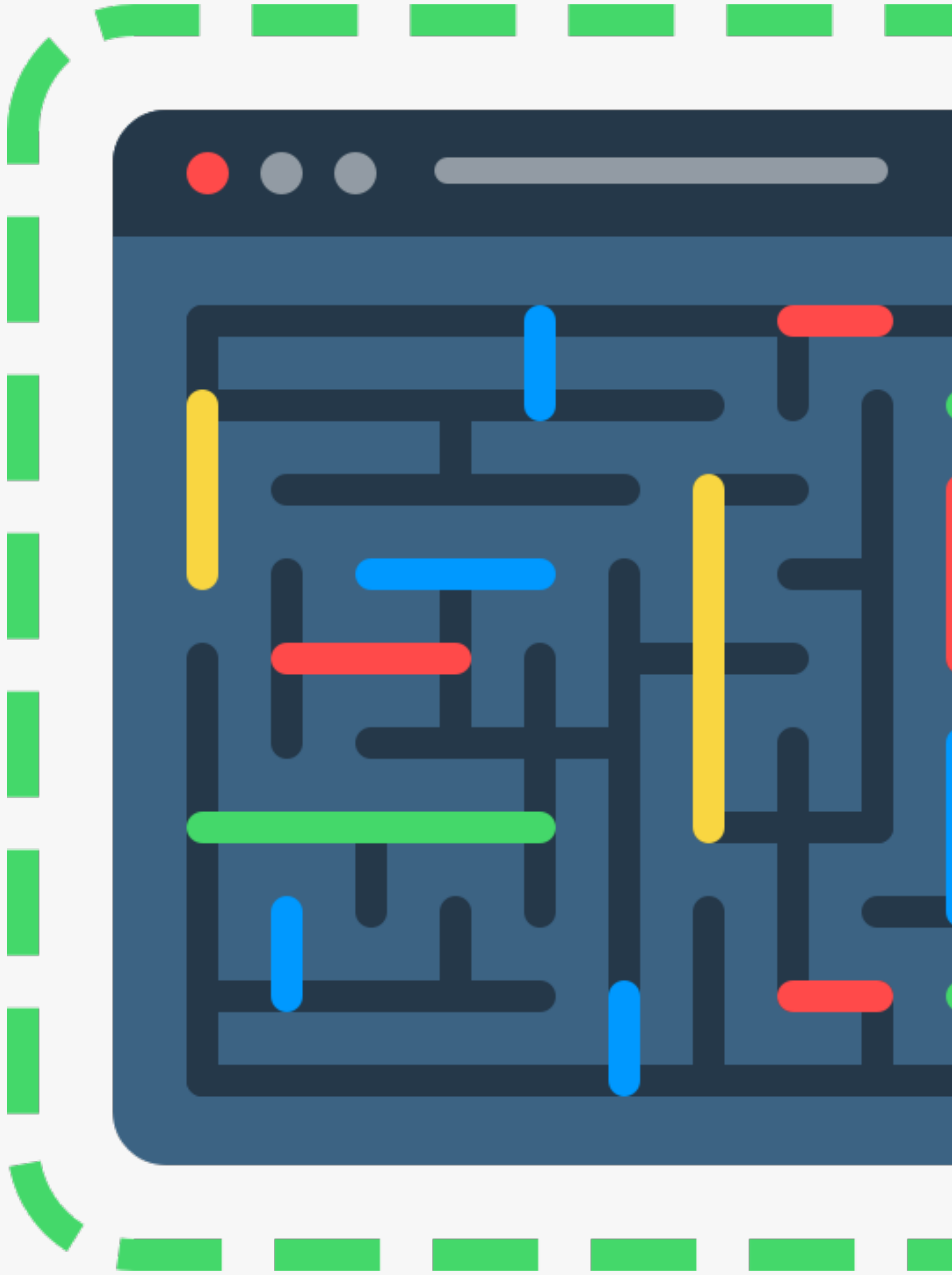


Control Transformations

- **Targets the control flow of the program**

- **Break up computations that logically belong together or merge computations that do not**
  - e.g. Function Outlining, Function Inlining, interleaved functions, cloned functions
- **Insert new code (redundant or dead) or make algorithmic changes**
- **Changes the ordering of functions and statements (changes locality of computations)**
- **Loop transformations - blocking / unrolling / fission**

- **Usually the most potent and resilient transformations**
- **Impact on performance is unavoidable**
- **Tradeoff between efficiency and obfuscation**

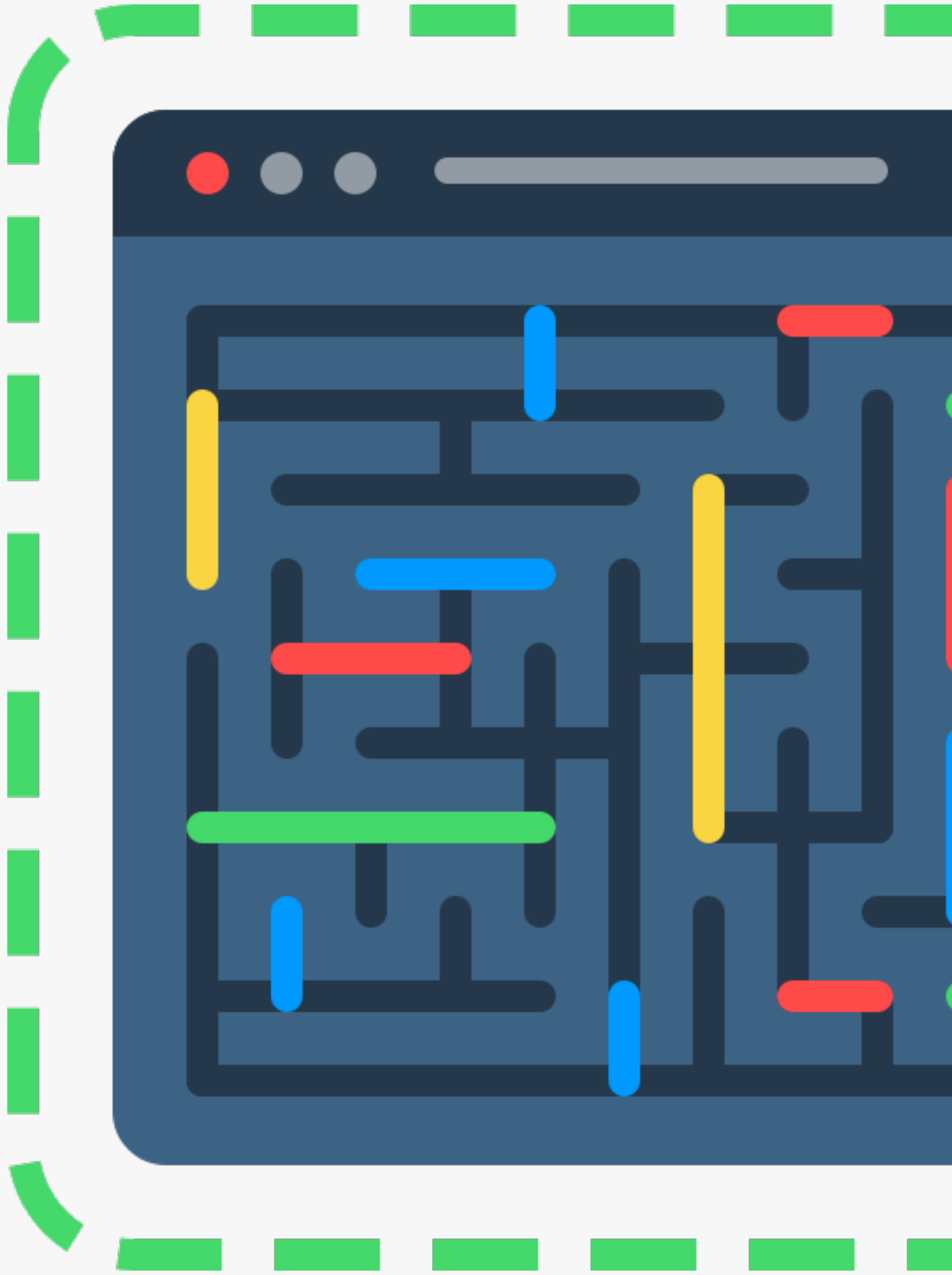


Data Transformations

- **Targets data structures**

- **Store data in unnatural storage classes**
  - e.g. store char literals in integers
- **Encoding**
- **Split-variables**
- **Function outlining (e.g. of a string generation)**
- **Array restructuring (split, merge, fold, flatten)**
- **Array shuffle**

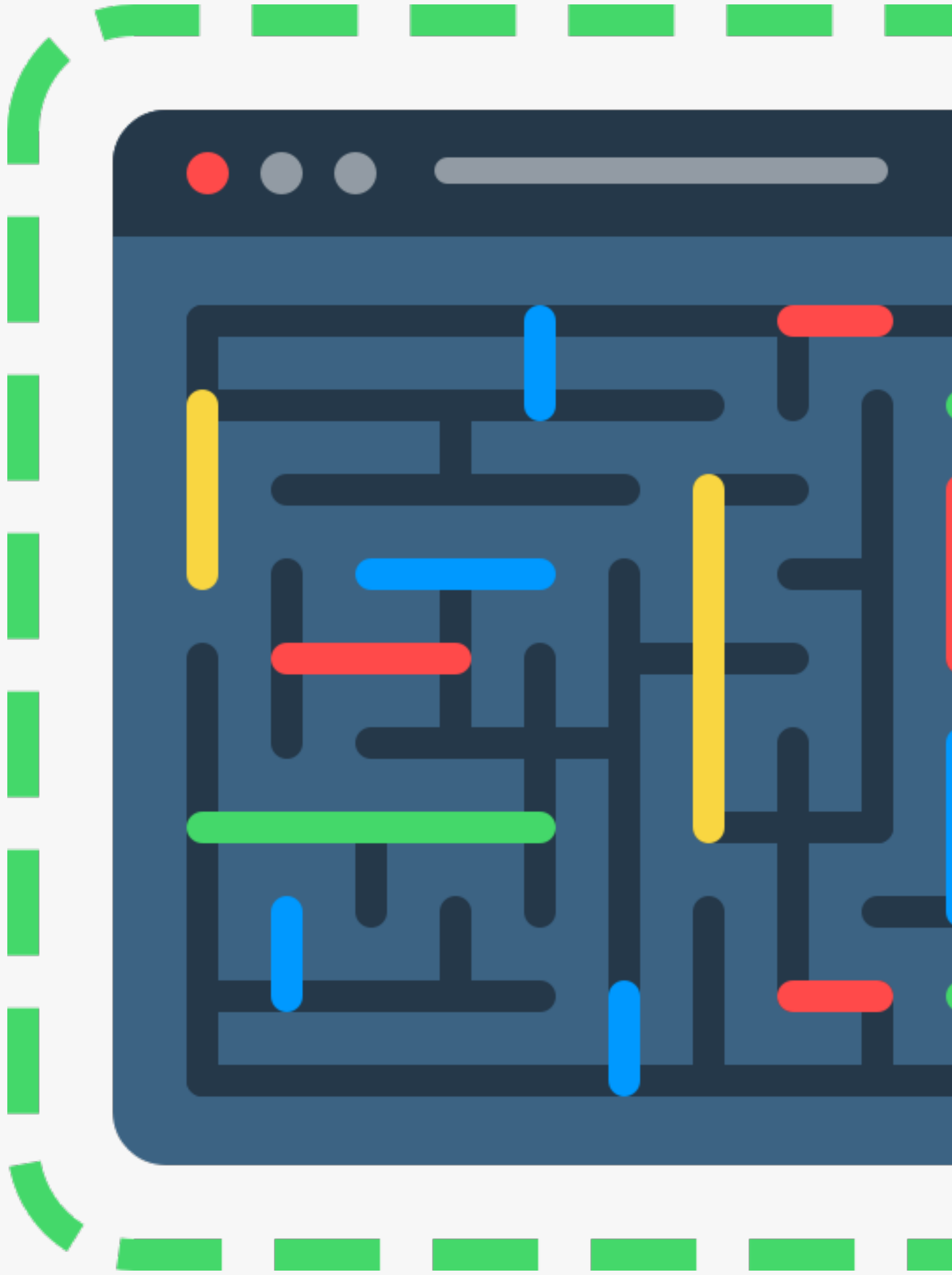
- **Suitable for temporary secrets or for stealthiness**



Preventive Transformations

- **Designed to reduce the efficiency of known obfuscation techniques and tools**

- **Examples:**
  - Add data dependencies to prevent automated reversal
  - Add number of variables to make automated tools become extremely slow and perhaps even crash
  - Explore know bugs in known reverse engineering tools
  - Add aliases and variable dependencies to preclude program slicing
  - Use of strong opaque predicates

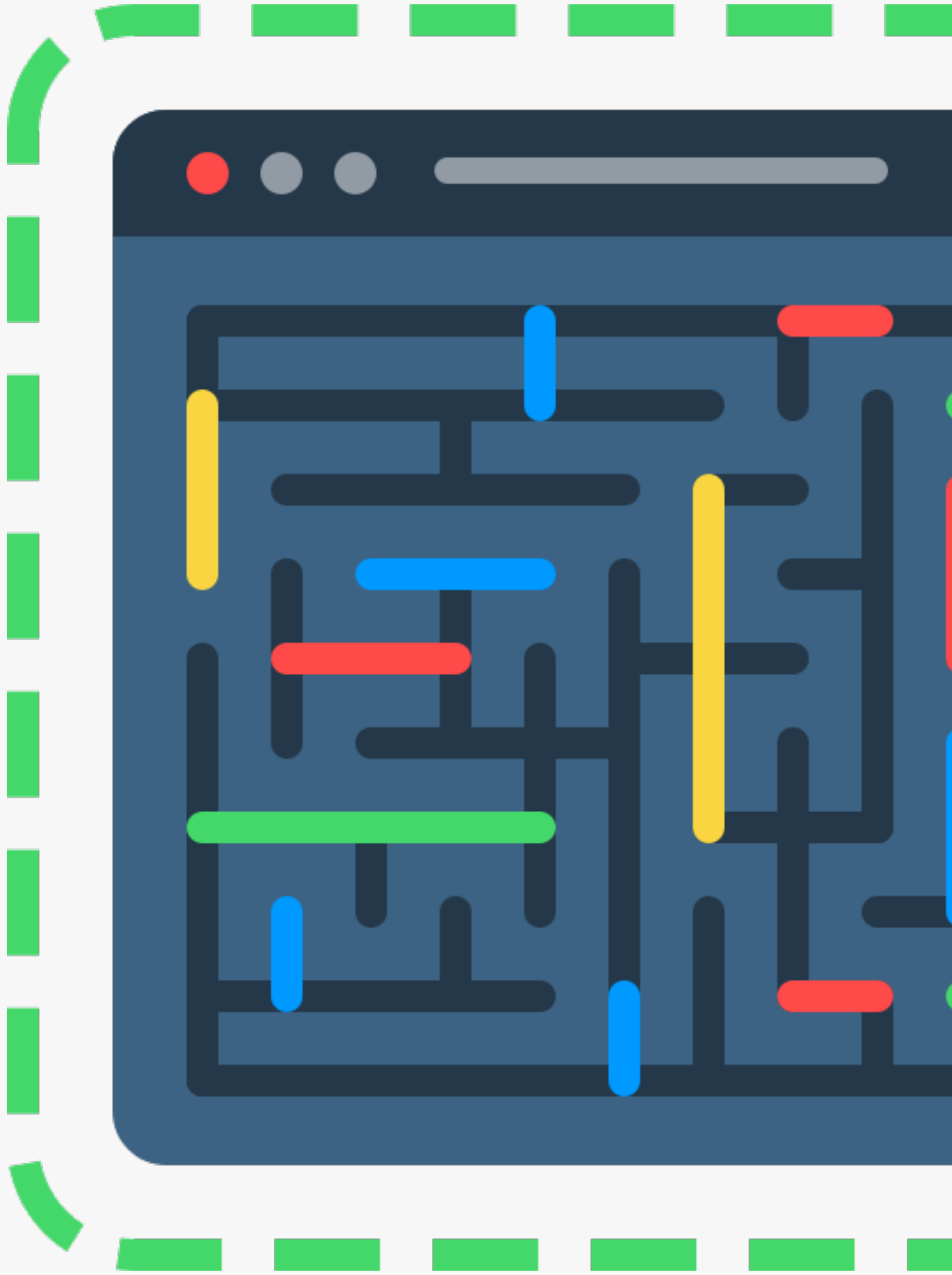


Opaque Predicates

- **Expression which is known to the obfuscator in compile time but difficult/costly for an automated debofuscator to revert**

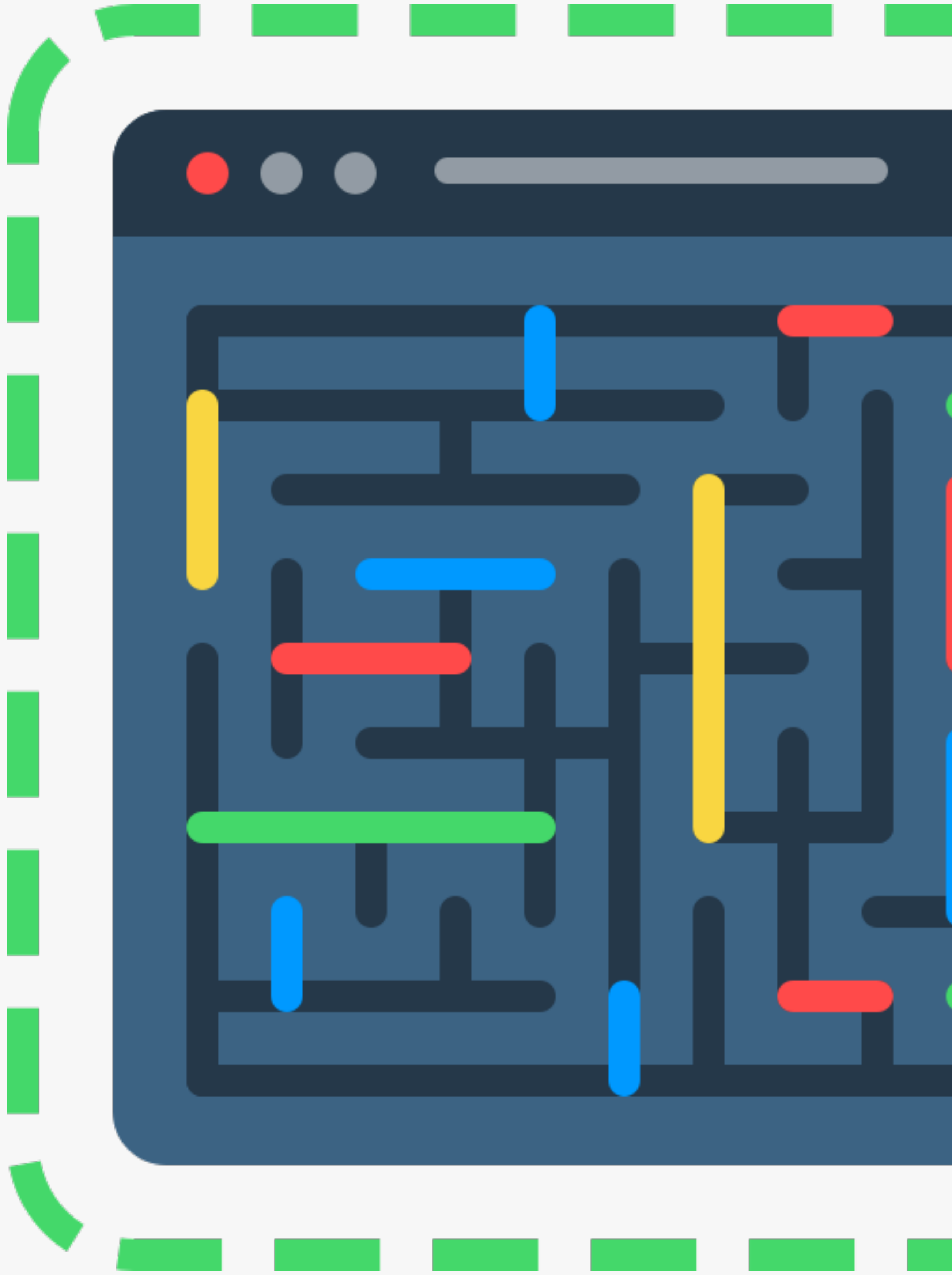
- **Examples:**
    - if (isPrime(15460178913505..1243)) ...
    - if (hashDigest(“string”) === “AB40...DFF”) ...
    - if (a * (a + 1) * a % 2 == 0) ...
    - if (a.b(c, d) !== e) ...

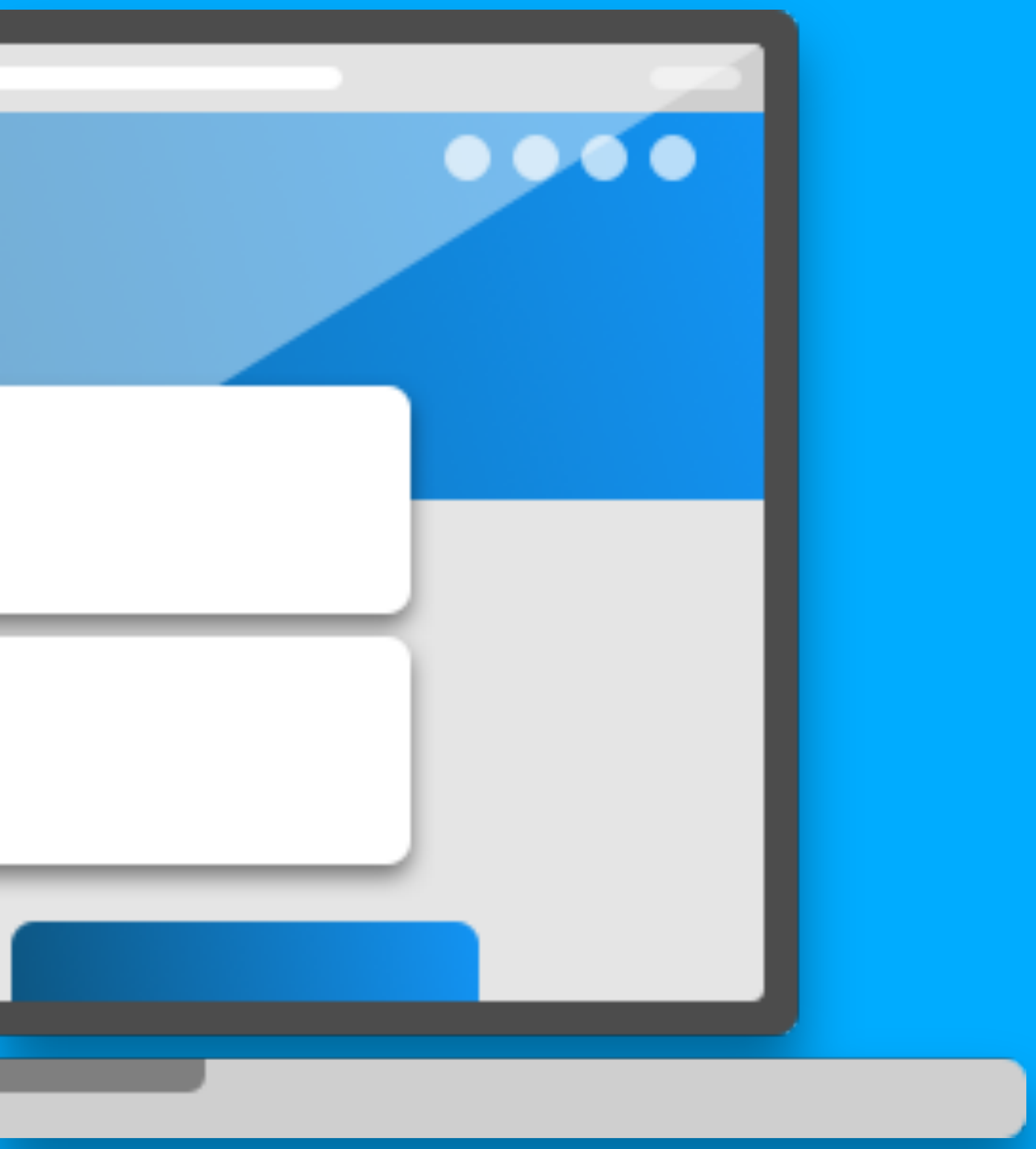
- **Shouldn’t be canned opaque predicates**
- **Ideally similar to real program constructs**
- **Deobfuscator tool can implement functions if they are predictable**
- **Essential for designing resilient control obfuscation transformations**
- **It’s not trivial to create highly resilient opaque predicates**



Scope of Transformation

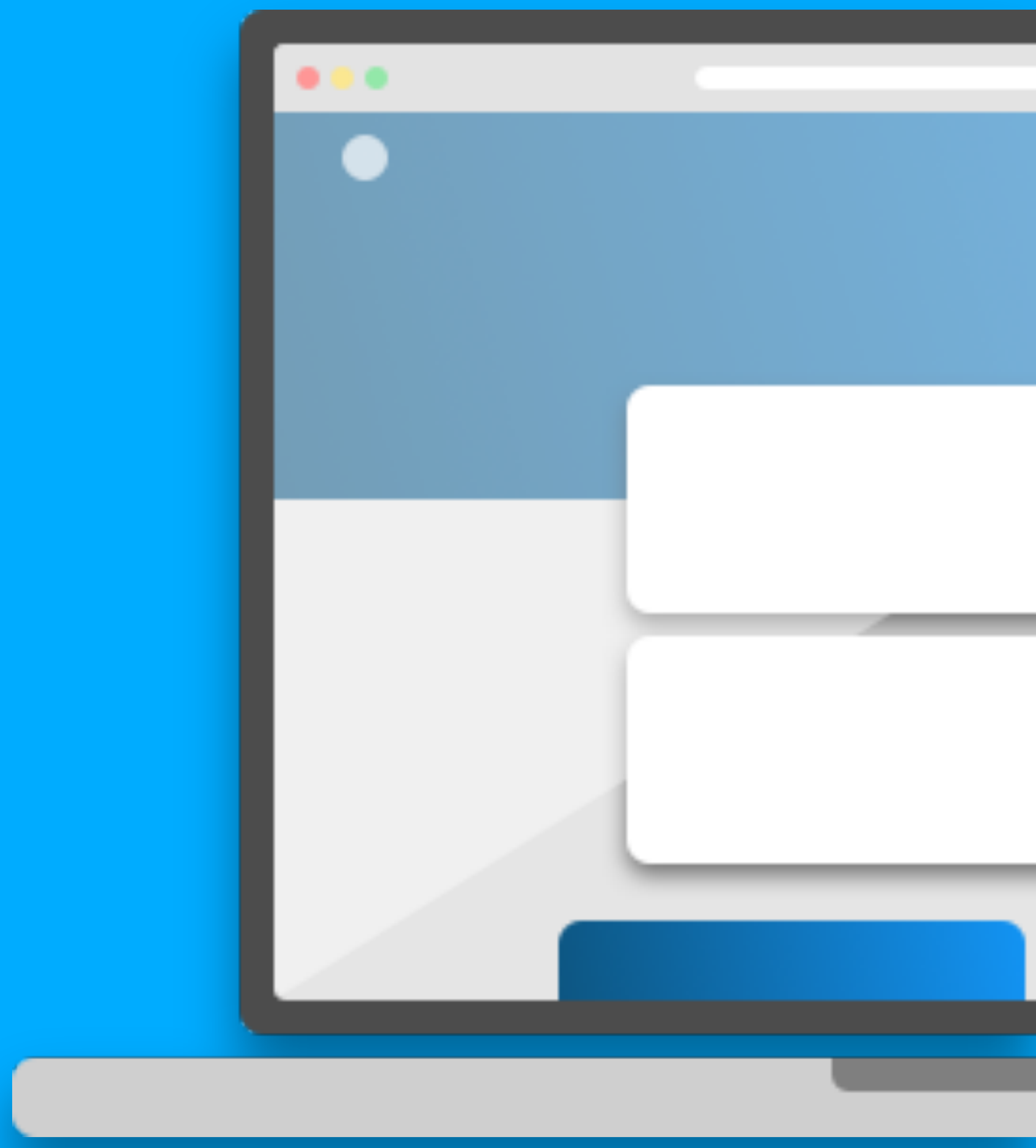
- **Local:** single basic block of a Control Flow Graph (CFG)
- **Global:** affects an entire CFG
- **Inter-procedural:** affects the flow of information between procedures
- **Inter-process:** affects the interaction between independently executing threads





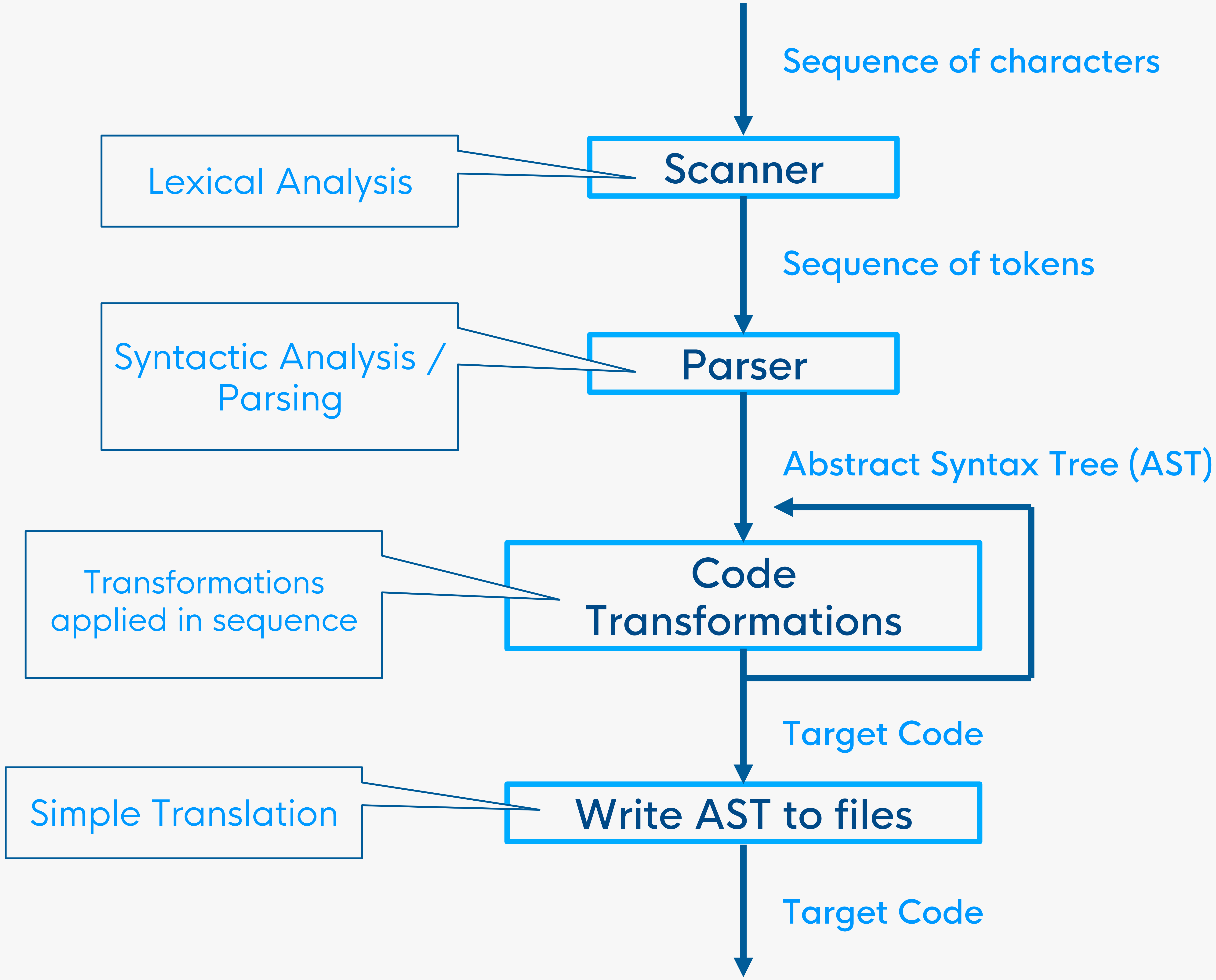
CODE OBFUSCATION PROCESS

PART 3

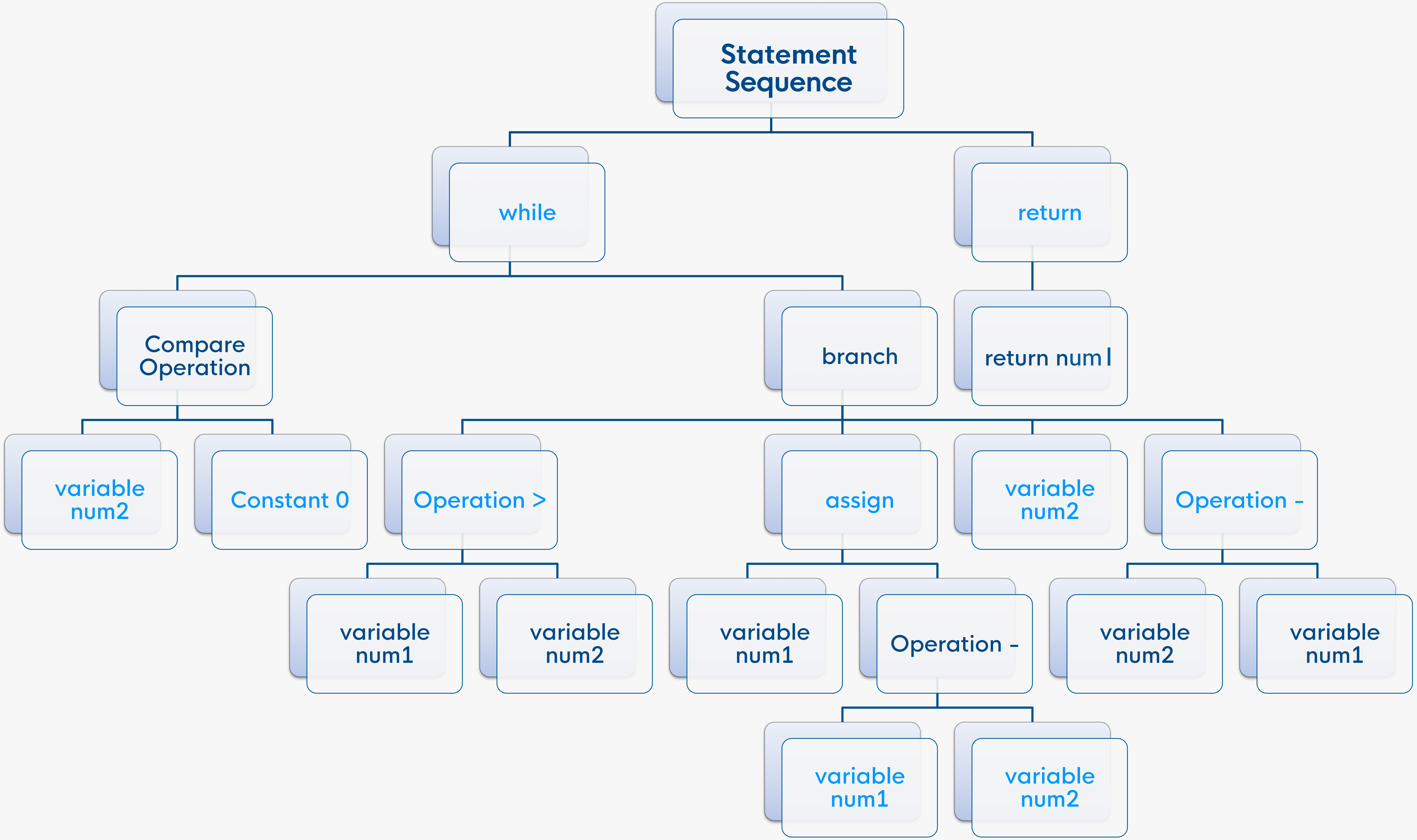


jscrambler

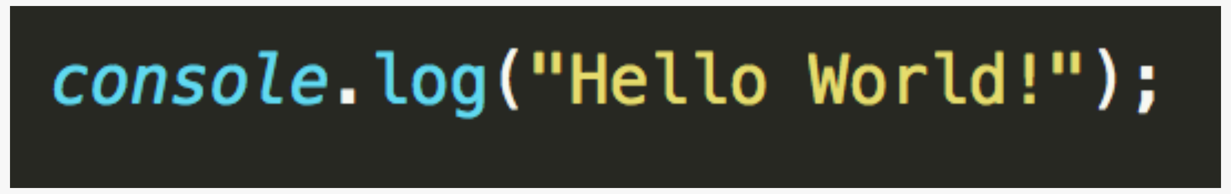
Code Transformation Process



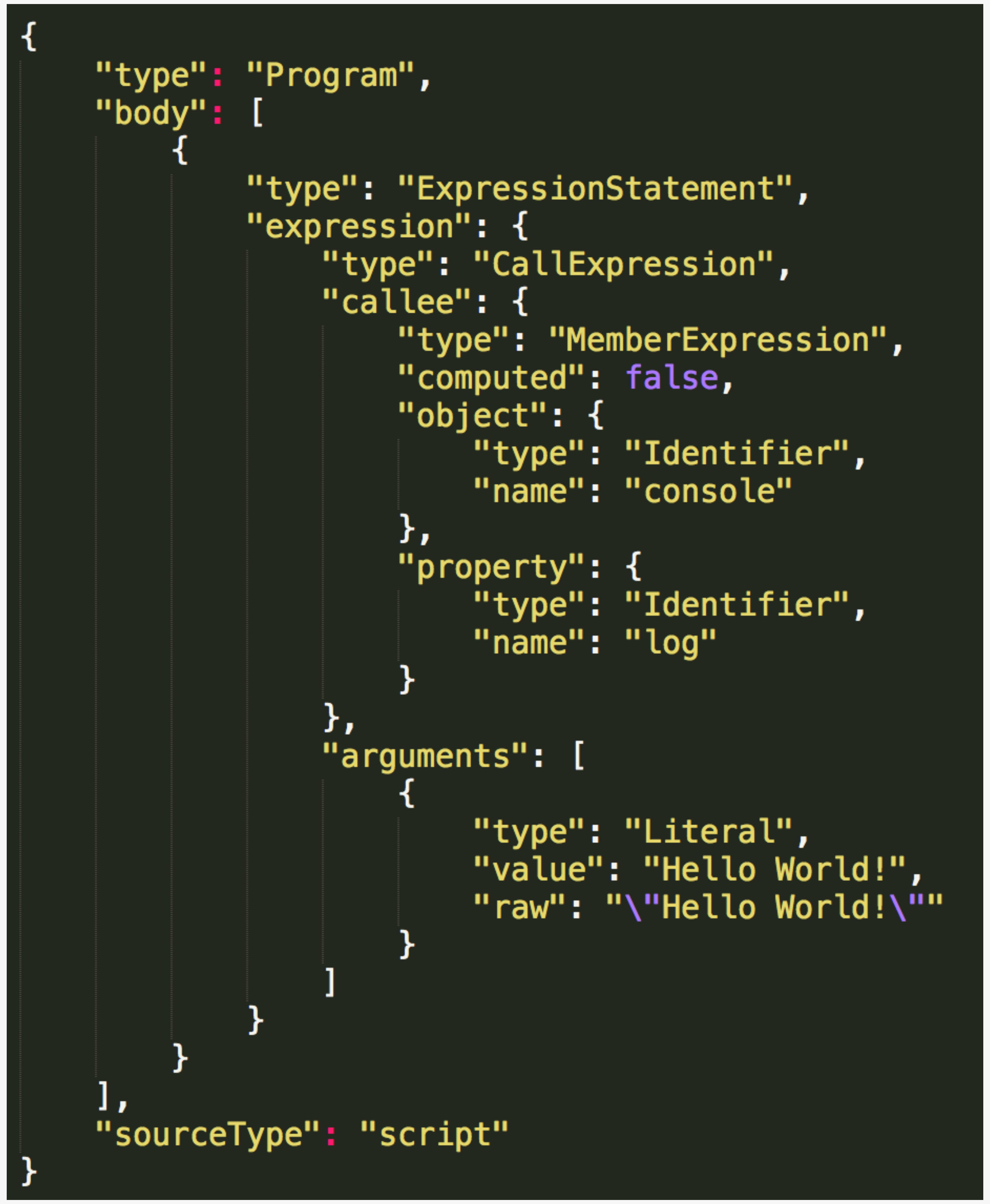
Abstract Syntax Tree



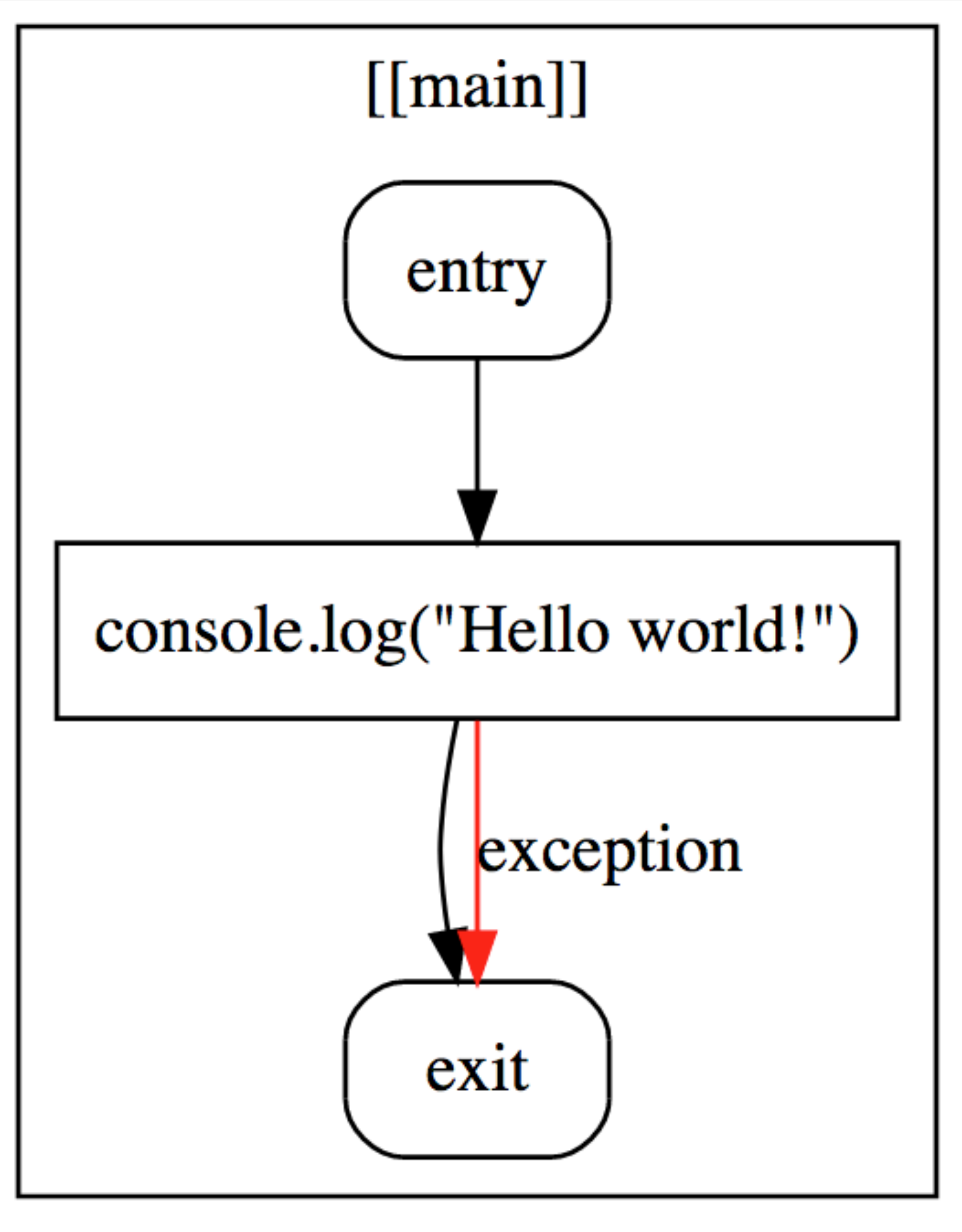
Abstract Syntax Tree



```
console.log("Hello World!");
```



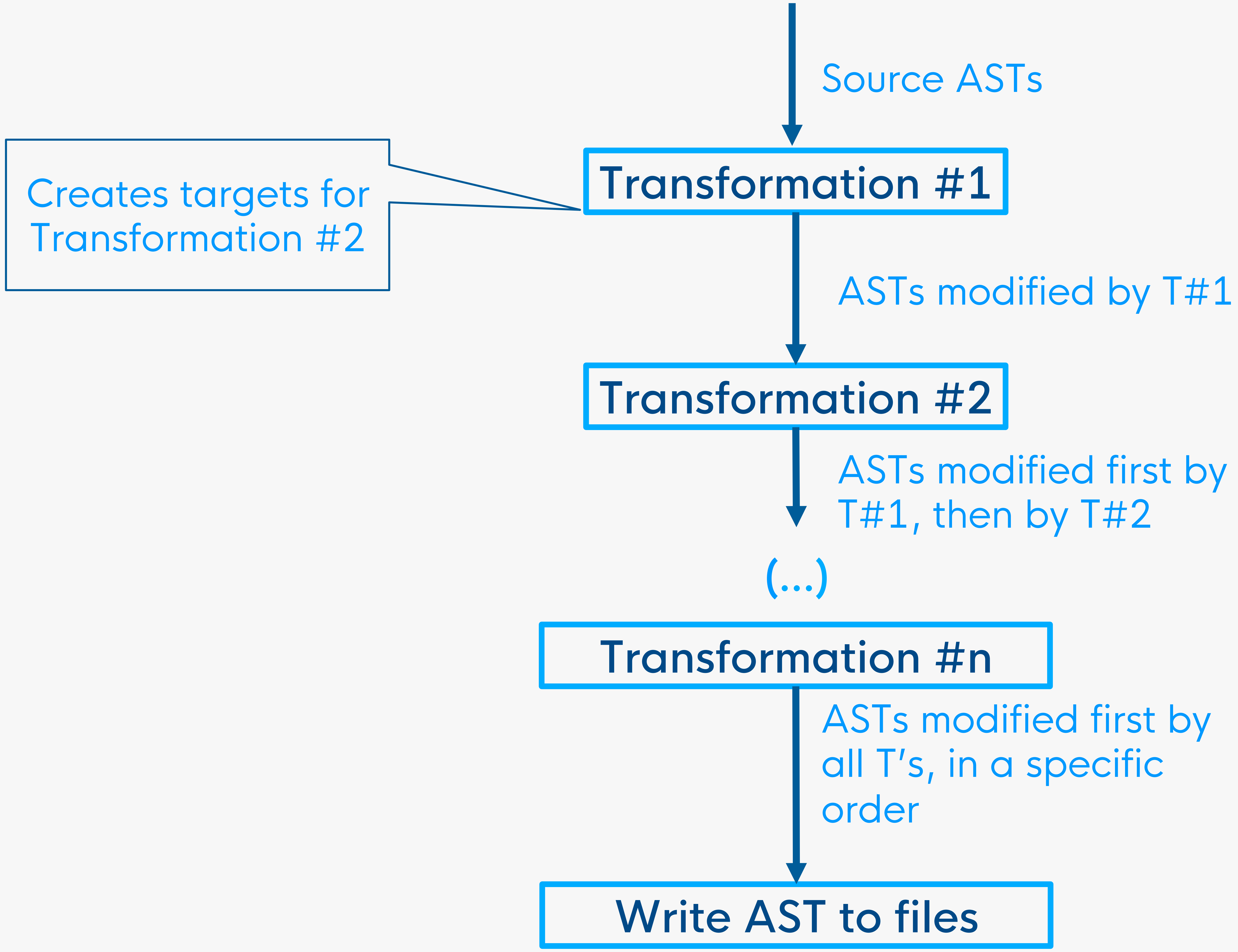
```
{
    "type": "Program",
    "body": [
        {
            "type": "ExpressionStatement",
            "expression": {
                "type": "CallExpression",
                "callee": {
                    "type": "MemberExpression",
                    "computed": false,
                    "object": {
                        "type": "Identifier",
                        "name": "console"
                    },
                    "property": {
                        "type": "Identifier",
                        "name": "log"
                    }
                },
                "arguments": [
                    {
                        "type": "Literal",
                        "value": "Hello World!",
                        "raw": "\"Hello World!\""
                    }
                ]
            }
        }
    ],
    "sourceType": "script"
}
```



http://esprima.org/demo/parse.html

http://azu.github.io/esgraph-graphviz-online/

Transformation Chaining effect


Source ASTs
Transformation #1
Creates targets for Transformation #2
ASTs modified by T#1
Transformation #2
ASTs modified first by T#1, then by T#2
(...)
Transformation #n
ASTs modified first by all T's, in a specific order
Write AST to files

- **Each transformation potentiates the ones that follow**
- **Order matters**
- **Randomizing order**
  - Higher diversity
  - Probably higher cost
- **Careful selection is advised**
  - Use good standards
  - Optionally, check with an expert

CODE OBFUSCATION TRANSFORMATIONS

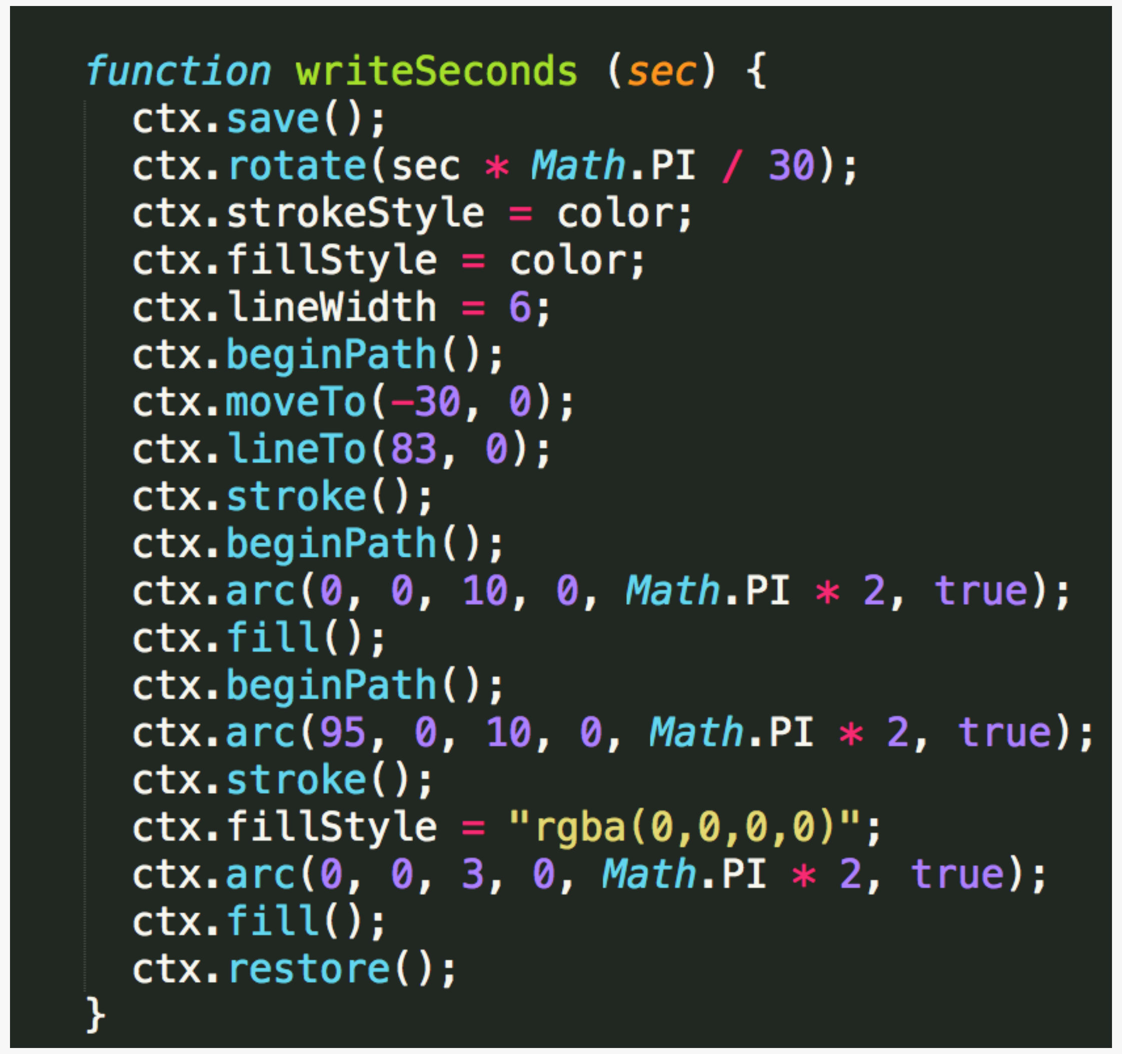
PART 4

jscrambler

Transformation Example #1

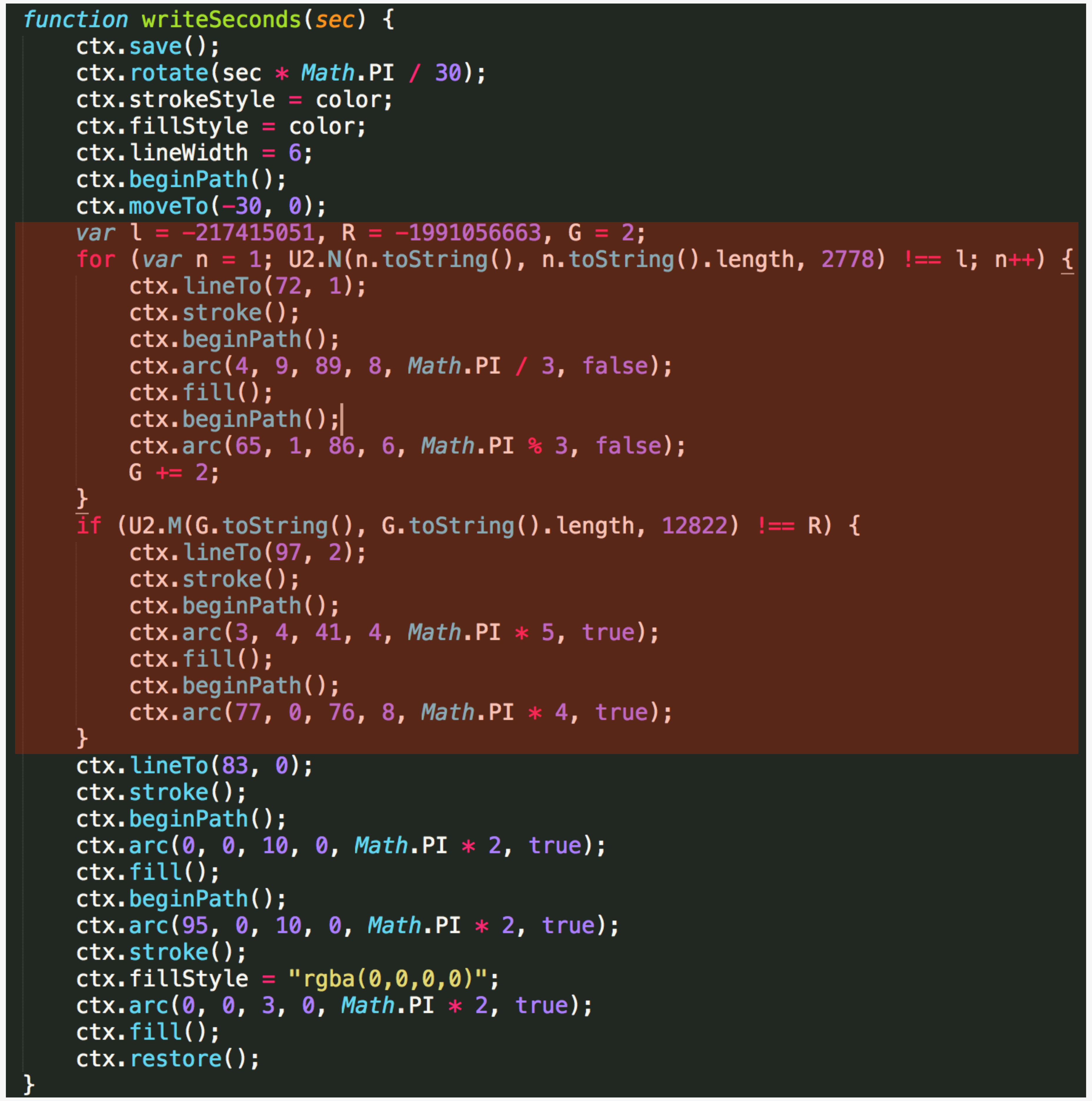
Dead code injection

- Generates statements similar to what exists in the program
- Uses strong non-local opaque predicates
- Cheap



```
function writeSeconds (sec) {
  ctx.save();
  ctx.rotate(sec * Math.PI / 30);
  ctx.strokeStyle = color;
  ctx.fillStyle = color;
  ctx.lineWidth = 6;
  ctx.beginPath();
  ctx.moveTo(-30, 0);
  ctx.lineTo(83, 0);
  ctx.stroke();
  ctx.beginPath();
  ctx.arc(0, 0, 10, 0, Math.PI * 2, true);
  ctx.fill();
  ctx.beginPath();
  ctx.arc(95, 0, 10, 0, Math.PI * 2, true);
  ctx.stroke();
  ctx.fillStyle = "rgba(0,0,0,0)";
  ctx.arc(0, 0, 3, 0, Math.PI * 2, true);
  ctx.fill();
  ctx.restore();
}
```

Dead code Injection

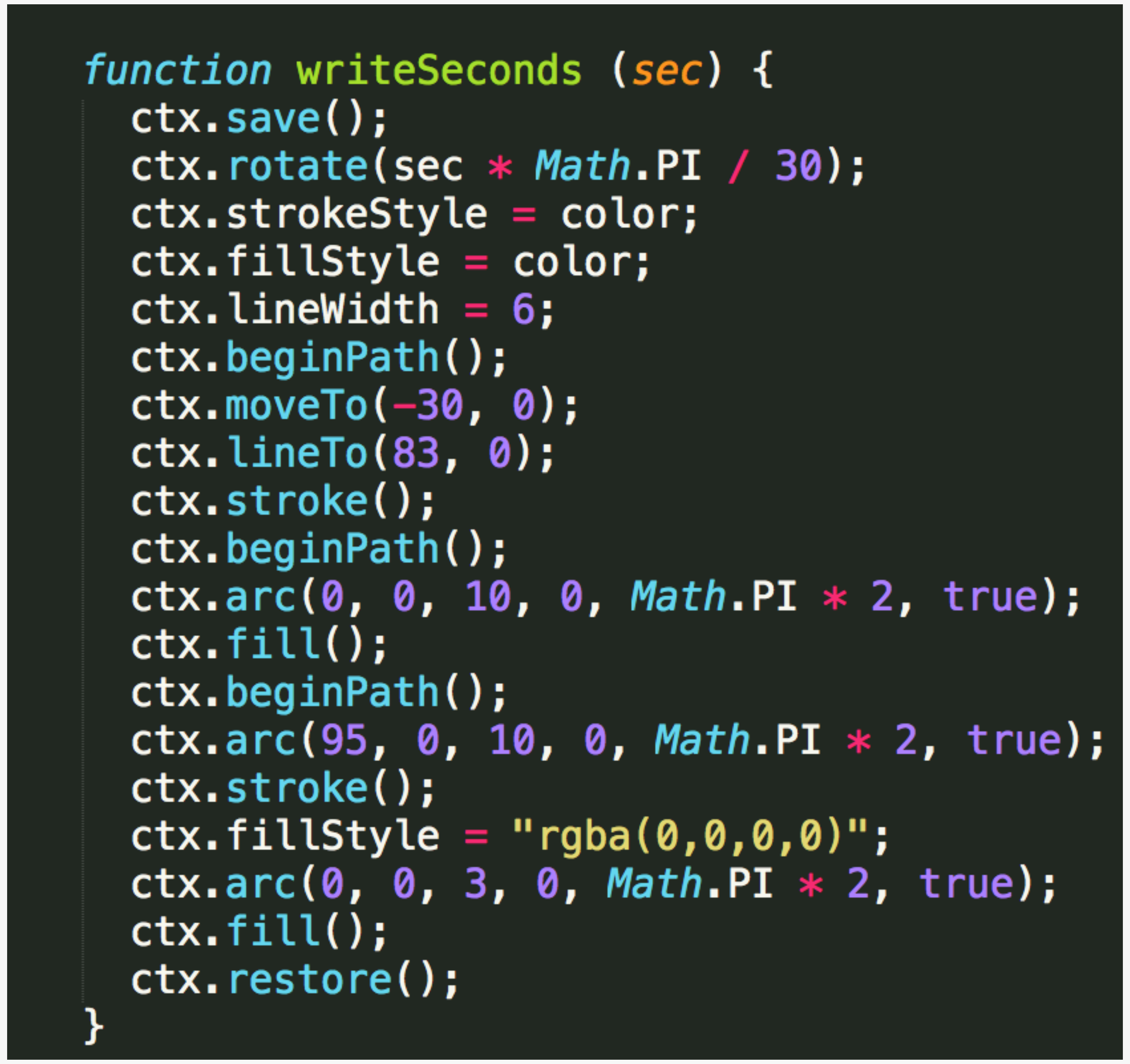


```
function writeSeconds(sec) {
    ctx.save();
    ctx.rotate(sec * Math.PI / 30);
    ctx.strokeStyle = color;
    ctx.fillStyle = color;
    ctx.lineWidth = 6;
    ctx.beginPath();
    ctx.moveTo(-30, 0);
    var l = -217415051, R = -1991056663, G = 2;
    for (var n = 1; U2.N(n.toString(), n.toString().length, 2778) !== l; n++) {
        ctx.lineTo(72, 1);
        ctx.stroke();
        ctx.beginPath();
        ctx.arc(4, 9, 89, 8, Math.PI / 3, false);
        ctx.fill();
        ctx.beginPath();
        ctx.arc(65, 1, 86, 6, Math.PI % 3, false);
        G += 2;
    }
    if (U2.M(G.toString(), G.toString().length, 12822) !== R) {
        ctx.lineTo(97, 2);
        ctx.stroke();
        ctx.beginPath();
        ctx.arc(3, 4, 41, 4, Math.PI * 5, true);
        ctx.fill();
        ctx.beginPath();
        ctx.arc(77, 0, 76, 8, Math.PI * 4, true);
    }
    ctx.lineTo(83, 0);
    ctx.stroke();
    ctx.beginPath();
    ctx.arc(0, 0, 10, 0, Math.PI * 2, true);
    ctx.fill();
    ctx.beginPath();
    ctx.arc(95, 0, 10, 0, Math.PI * 2, true);
    ctx.stroke();
    ctx.fillStyle = "rgba(0,0,0,0)";
    ctx.arc(0, 0, 3, 0, Math.PI * 2, true);
    ctx.fill();
    ctx.restore();
}
```

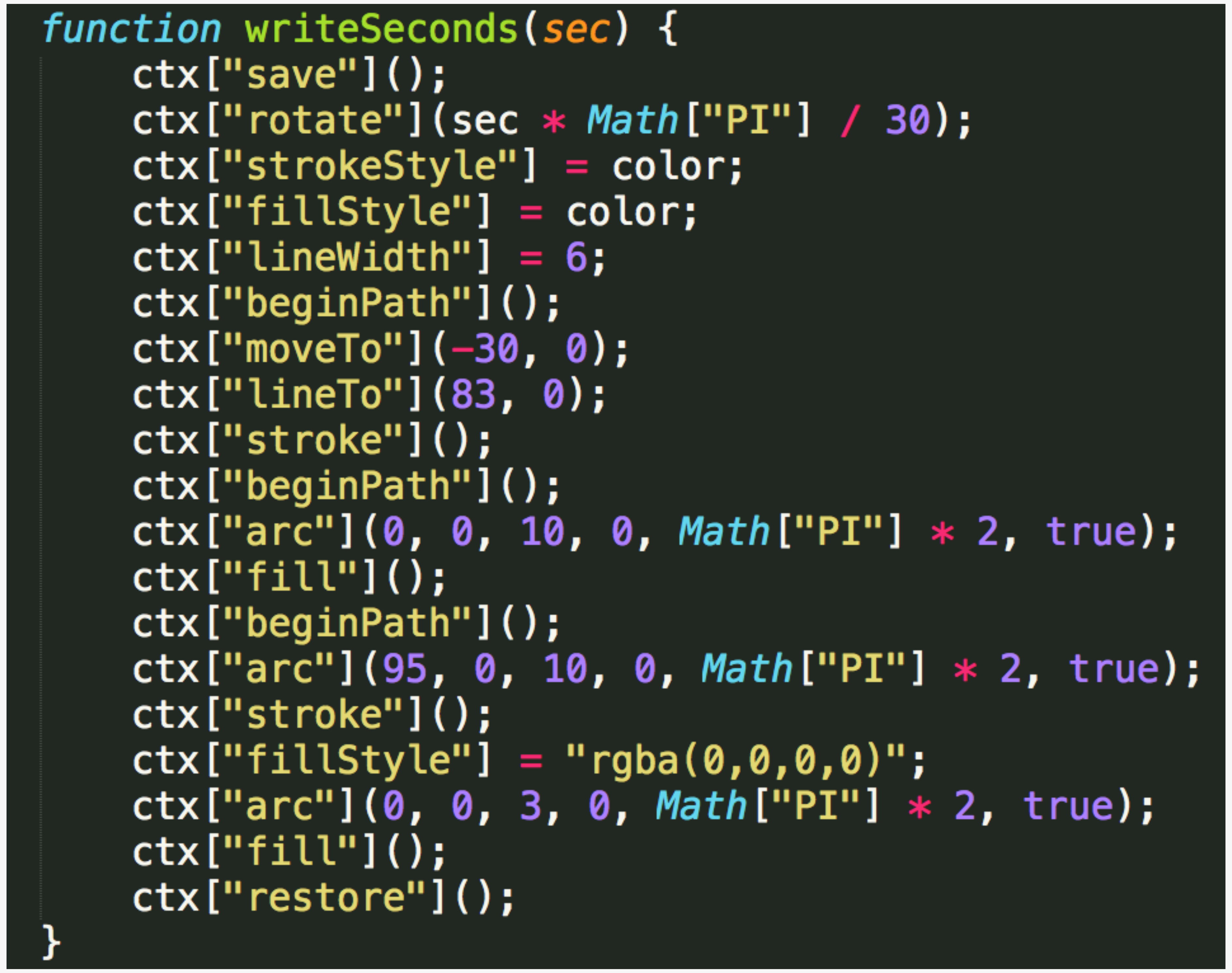
Transformation Example #2

Dot to bracket notation

- Zero potency, Zero Resiliency, cheap
- Why would we want this?



```
function writeSeconds (sec) {
  ctx.save();
  ctx.rotate(sec * Math.PI / 30);
  ctx.strokeStyle = color;
  ctx.fillStyle = color;
  ctx.lineWidth = 6;
  ctx.beginPath();
  ctx.moveTo(-30, 0);
  ctx.lineTo(83, 0);
  ctx.stroke();
  ctx.beginPath();
  ctx.arc(0, 0, 10, 0, Math.PI * 2, true);
  ctx.fill();
  ctx.beginPath();
  ctx.arc(95, 0, 10, 0, Math.PI * 2, true);
  ctx.stroke();
  ctx.fillStyle = "rgba(0,0,0,0)";
  ctx.arc(0, 0, 3, 0, Math.PI * 2, true);
  ctx.fill();
  ctx.restore();
}
```

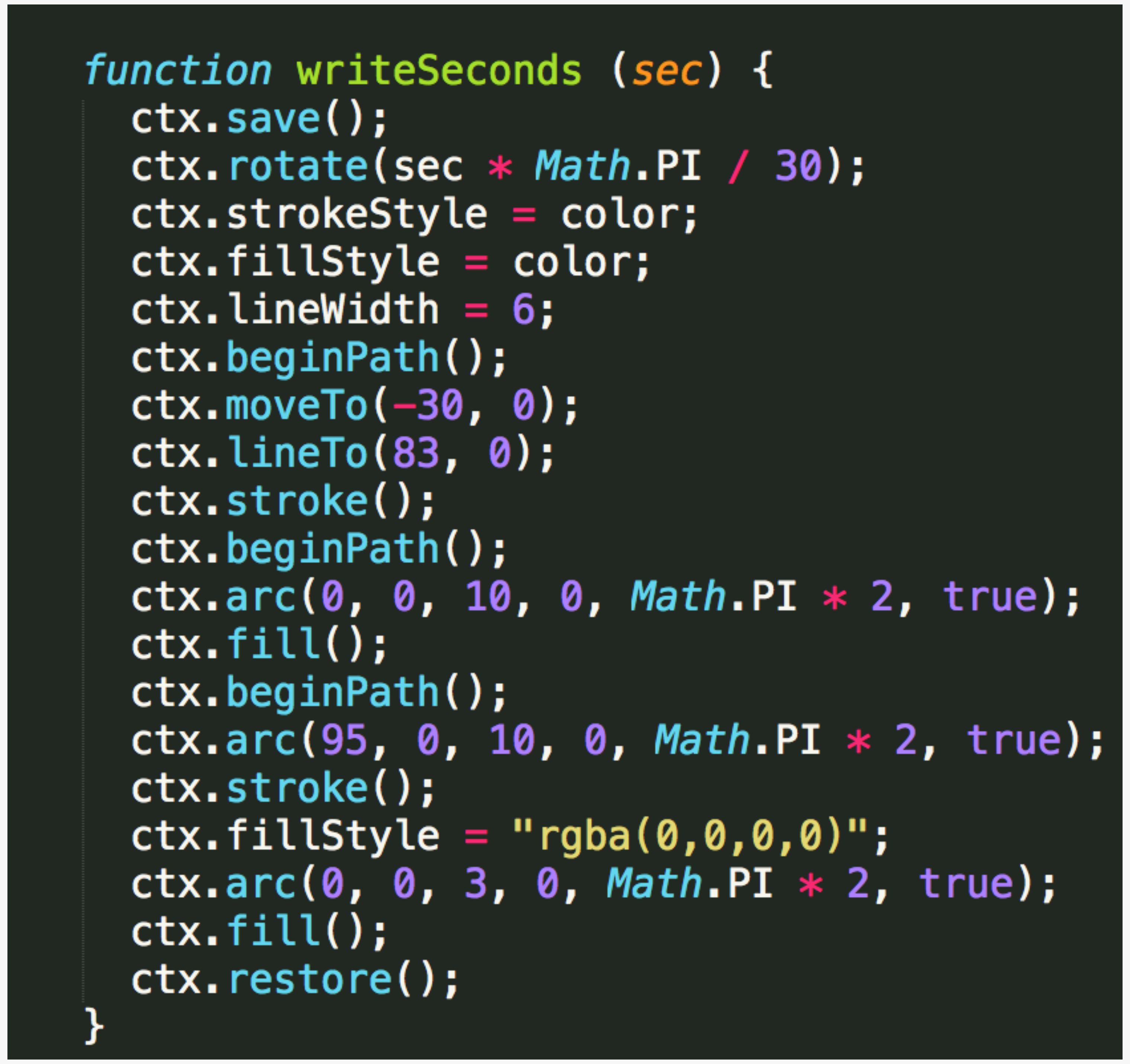


```
function writeSeconds(sec) {
    ctx["save"]();
    ctx["rotate"](sec * Math["PI"] / 30);
    ctx["strokeStyle"] = color;
    ctx["fillStyle"] = color;
    ctx["lineWidth"] = 6;
    ctx["beginPath"]();
    ctx["moveTo"](-30, 0);
    ctx["lineTo"](83, 0);
    ctx["stroke"]();
    ctx["beginPath"]();
    ctx["arc"](0, 0, 10, 0, Math["PI"] * 2, true);
    ctx["fill"]();
    ctx["beginPath"]();
    ctx["arc"](95, 0, 10, 0, Math["PI"] * 2, true);
    ctx["stroke"]();
    ctx["fillStyle"] = "rgba(0,0,0,0)";
    ctx["arc"](0, 0, 3, 0, Math["PI"] * 2, true);
    ctx["fill"]();
    ctx["restore"]();
}
```

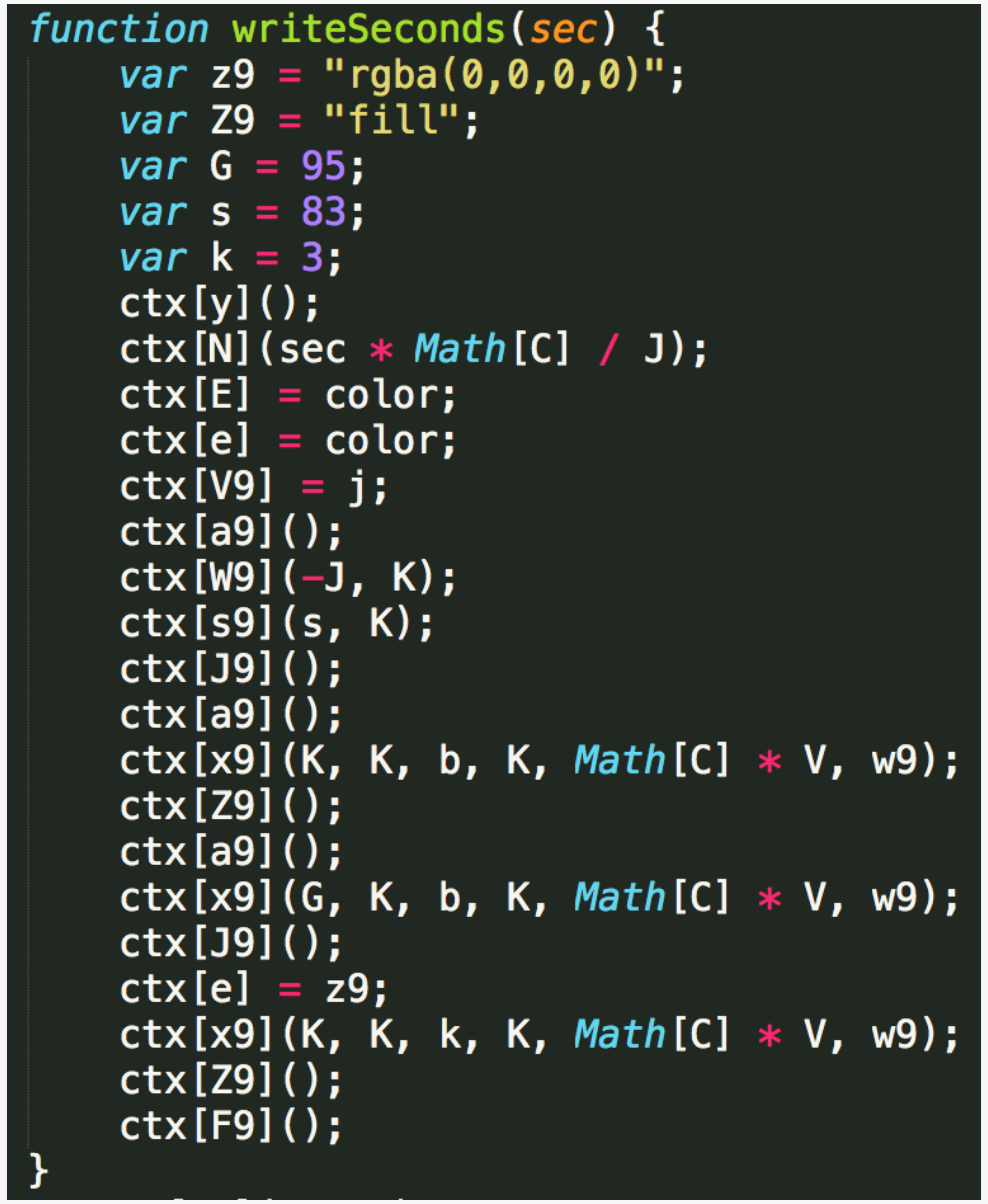
Transformation Example #3

Dot to bracket notation + Duplicate Literals Removal

- Generated more targets
- Some transformations are only meant to potentiate others



```
function writeSeconds (sec) {
  ctx.save();
  ctx.rotate(sec * Math.PI / 30);
  ctx.strokeStyle = color;
  ctx.fillStyle = color;
  ctx.lineWidth = 6;
  ctx.beginPath();
  ctx.moveTo(-30, 0);
  ctx.lineTo(83, 0);
  ctx.stroke();
  ctx.beginPath();
  ctx.arc(0, 0, 10, 0, Math.PI * 2, true);
  ctx.fill();
  ctx.beginPath();
  ctx.arc(95, 0, 10, 0, Math.PI * 2, true);
  ctx.stroke();
  ctx.fillStyle = "rgba(0,0,0,0)";
  ctx.arc(0, 0, 3, 0, Math.PI * 2, true);
  ctx.fill();
  ctx.restore();
}
```

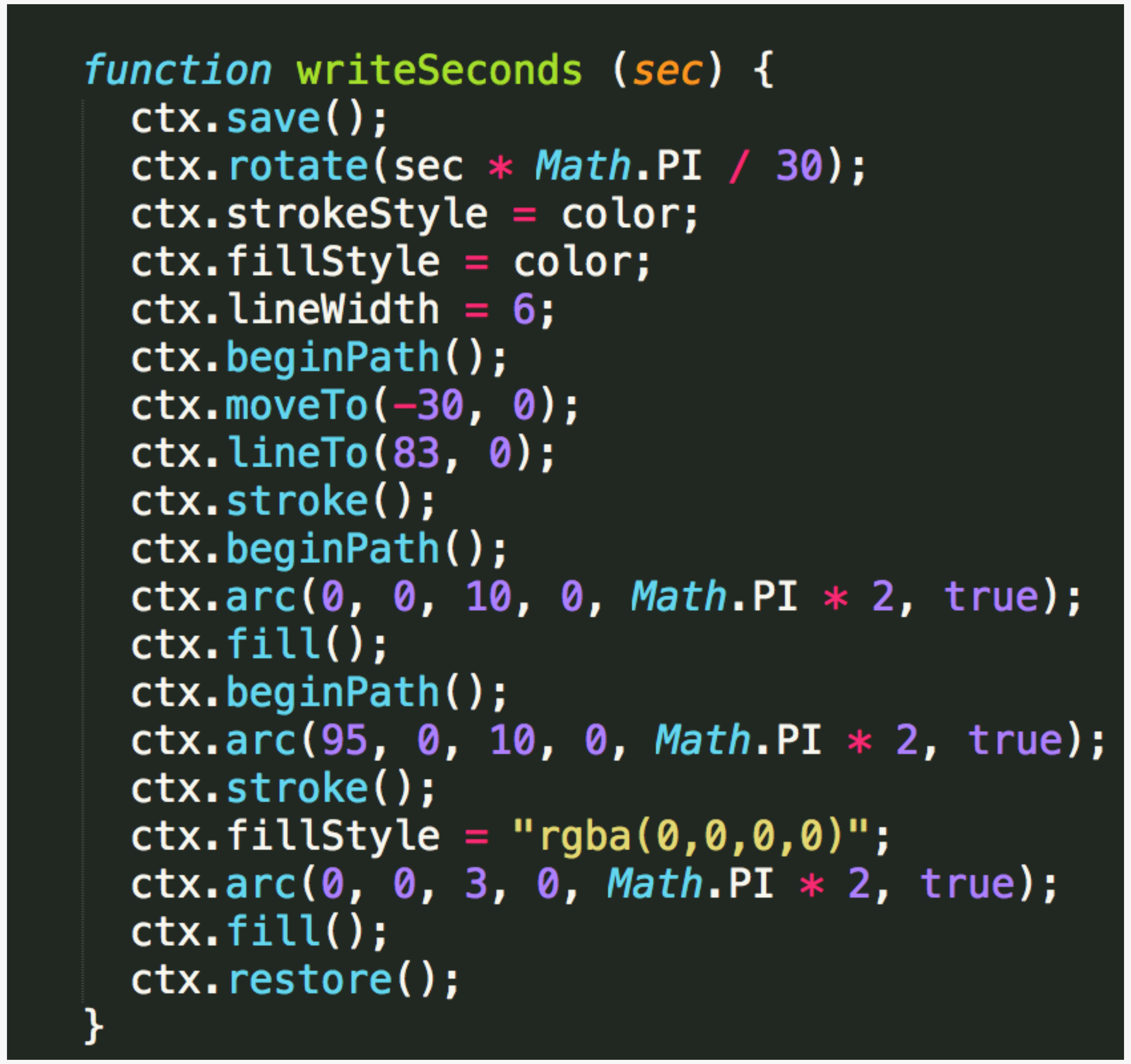


```
function writeSeconds(sec) {
    var z9 = "rgba(0,0,0,0)";
    var Z9 = "fill";
    var G = 95;
    var s = 83;
    var k = 3;
    ctx[y]();
    ctx[N](sec * Math[C] / J);
    ctx[E] = color;
    ctx[e] = color;
    ctx[V9] = j;
    ctx[a9]();
    ctx[W9](-J, K);
    ctx[s9](s, K);
    ctx[J9]();
    ctx[a9]();
    ctx[x9](K, K, b, K, Math[C] * V, w9);
    ctx[Z9]();
    ctx[a9]();
    ctx[x9](G, K, b, K, Math[C] * V, w9);
    ctx[J9]();
    ctx[e] = z9;
    ctx[x9](K, K, k, K, Math[C] * V, w9);
    ctx[Z9]();
    ctx[F9]();
}
```

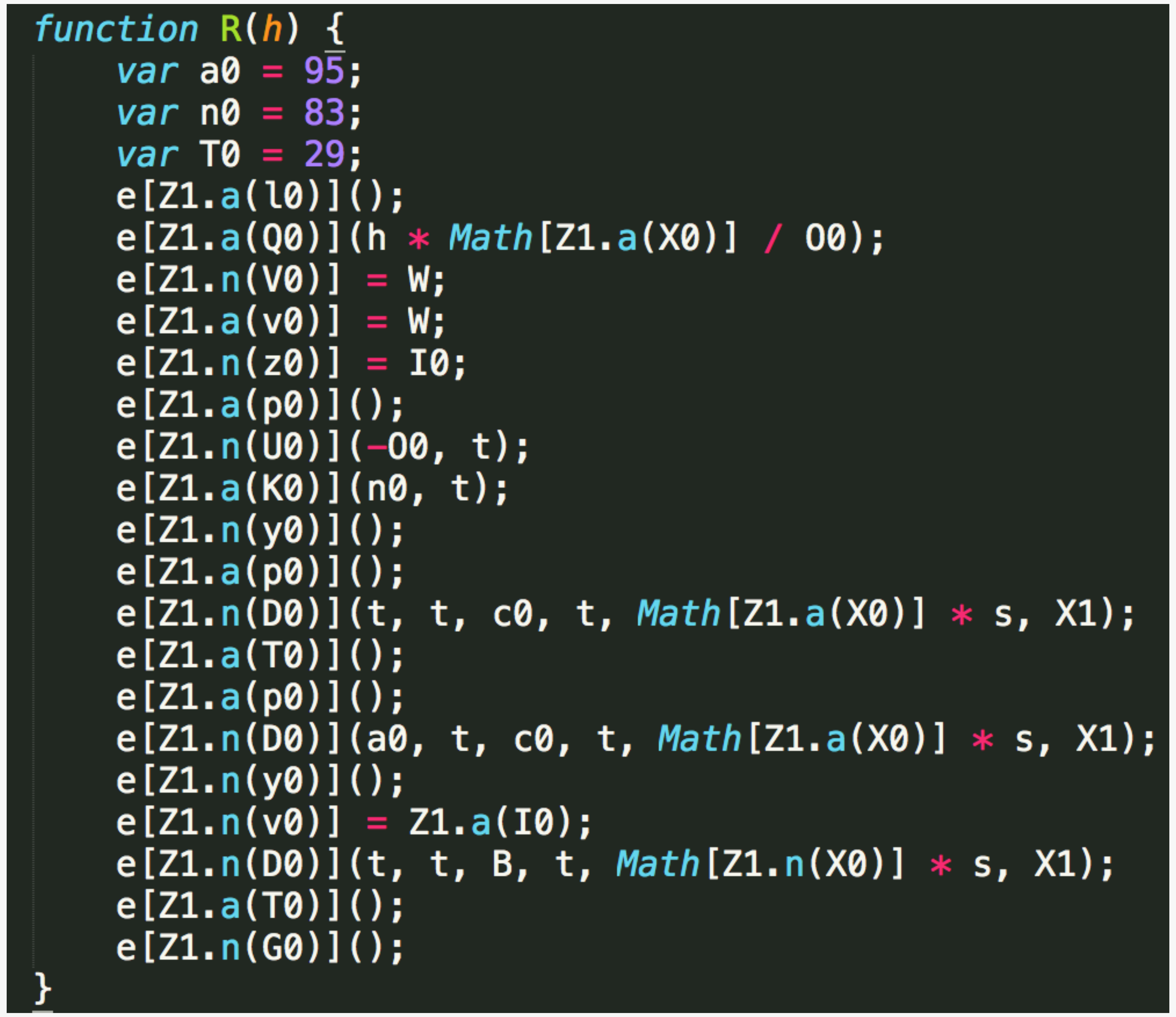
Transformation Example #4

Dot to bracket notation + Duplicate Literals Removal + String Splitting & Concealing + Identifiers Renaming

- **Eliminated strings and object names**
- **But we haven’t really changed the control flow that much**



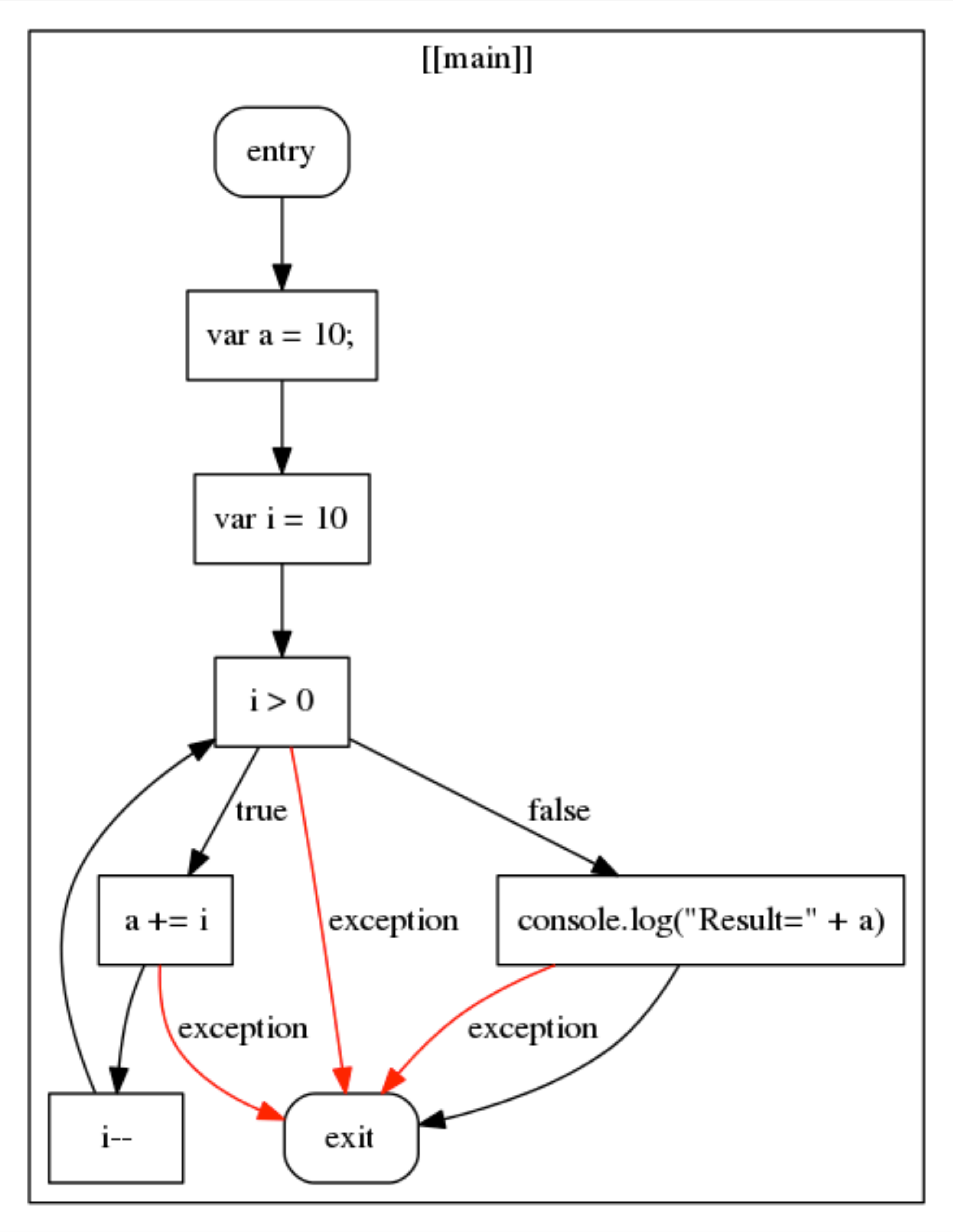
```
function writeSeconds (sec) {
   ctx.save();
   ctx.rotate(sec * Math.PI / 30);
   ctx.strokeStyle = color;
   ctx.fillStyle = color;
   ctx.lineWidth = 6;
   ctx.beginPath();
   ctx.moveTo(-30, 0);
   ctx.lineTo(83, 0);
   ctx.stroke();
   ctx.beginPath();
   ctx.arc(0, 0, 10, 0, Math.PI * 2, true);
   ctx.fill();
   ctx.beginPath();
   ctx.arc(95, 0, 10, 0, Math.PI * 2, true);
   ctx.stroke();
   ctx.fillStyle = "rgba(0,0,0,0)";
   ctx.arc(0, 0, 3, 0, Math.PI * 2, true);
   ctx.fill();
   ctx.restore();
}
```

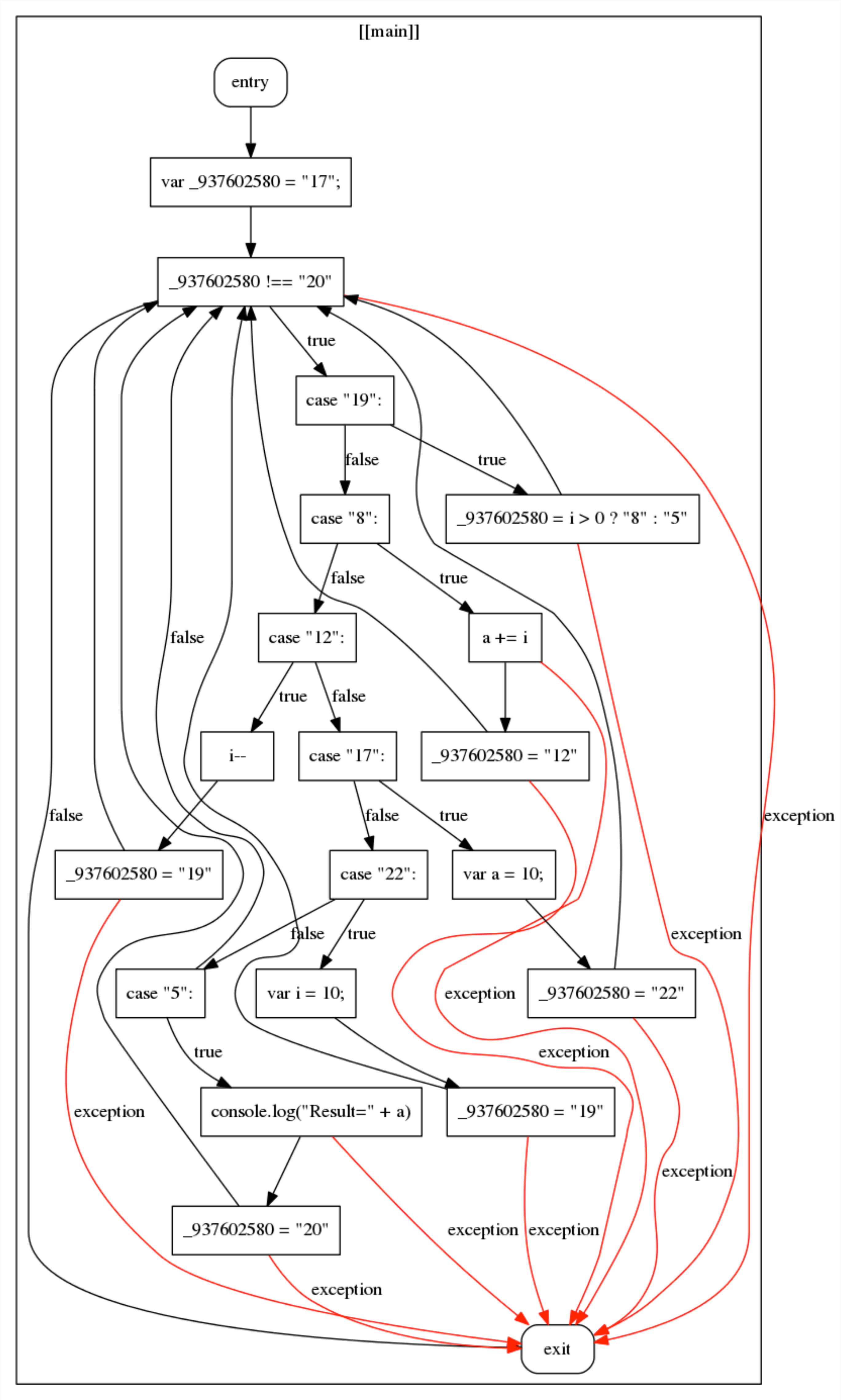


```
function R(h) {
    var a0 = 95;
    var n0 = 83;
    var T0 = 29;
    e[Z1.a(l0)]();
    e[Z1.a(Q0)](h * Math[Z1.a(X0)] / O0);
    e[Z1.n(V0)] = W;
    e[Z1.a(v0)] = W;
    e[Z1.n(z0)] = I0;
    e[Z1.a(p0)]();
    e[Z1.n(U0)](-O0, t);
    e[Z1.a(K0)](n0, t);
    e[Z1.n(y0)]();
    e[Z1.a(p0)]();
    e[Z1.n(D0)](t, t, c0, t, Math[Z1.a(X0)] * s, X1);
    e[Z1.a(T0)]();
    e[Z1.a(p0)]();
    e[Z1.n(D0)](a0, t, c0, t, Math[Z1.a(X0)] * s, X1);
    e[Z1.n(y0)]();
    e[Z1.n(v0)] = Z1.a(I0);
    e[Z1.n(D0)](t, t, B, t, Math[Z1.n(X0)] * s, X1);
    e[Z1.a(T0)]();
    e[Z1.n(G0)]();
}
```

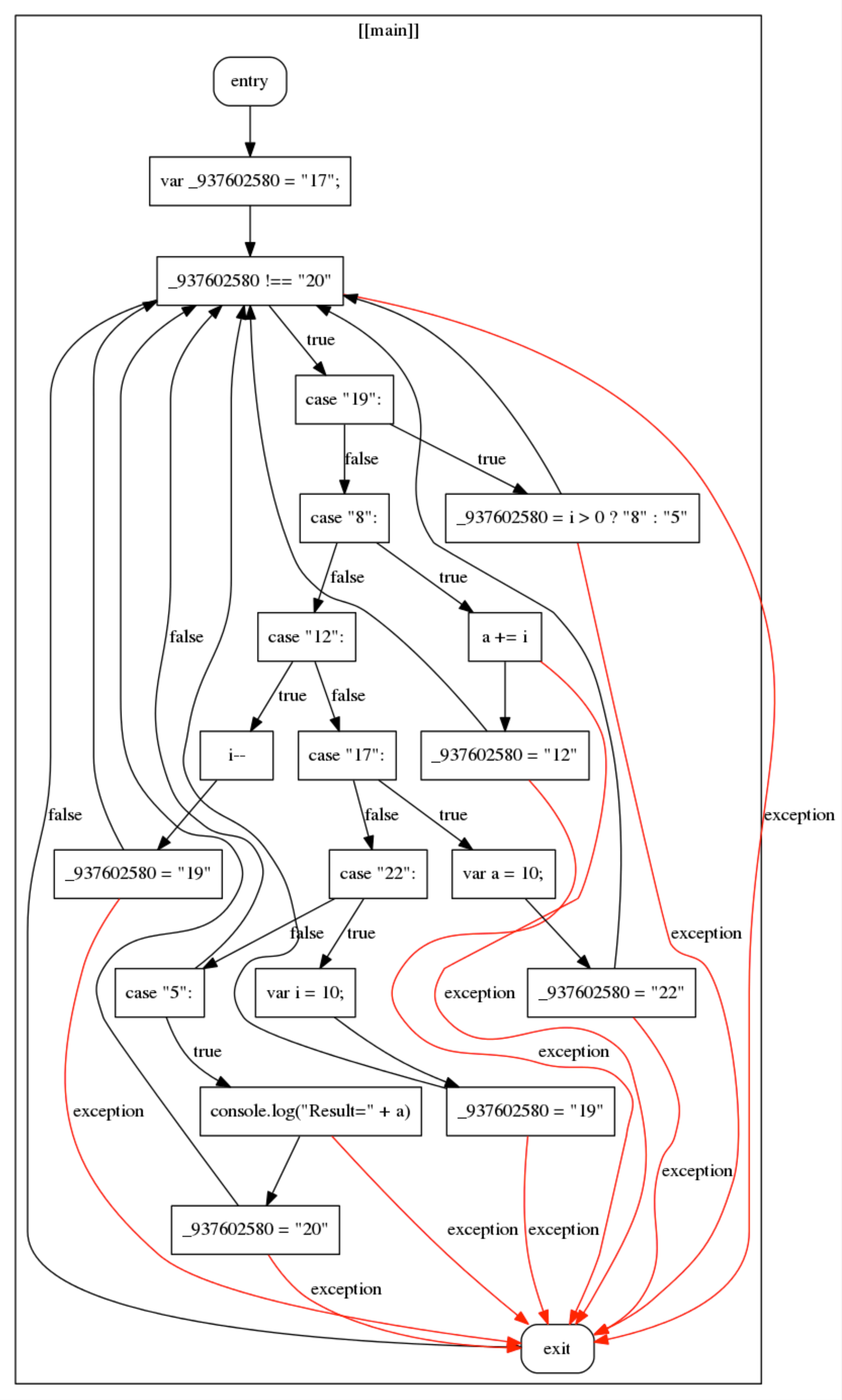
Control Flow Flattening

- Splits all the source code basic blocks and puts them all inside a single infinite loop with a `switch` statement that controls the program flow
- program flow becomes significantly harder to follow because natural conditional constructs that made the code easier to read are now gone

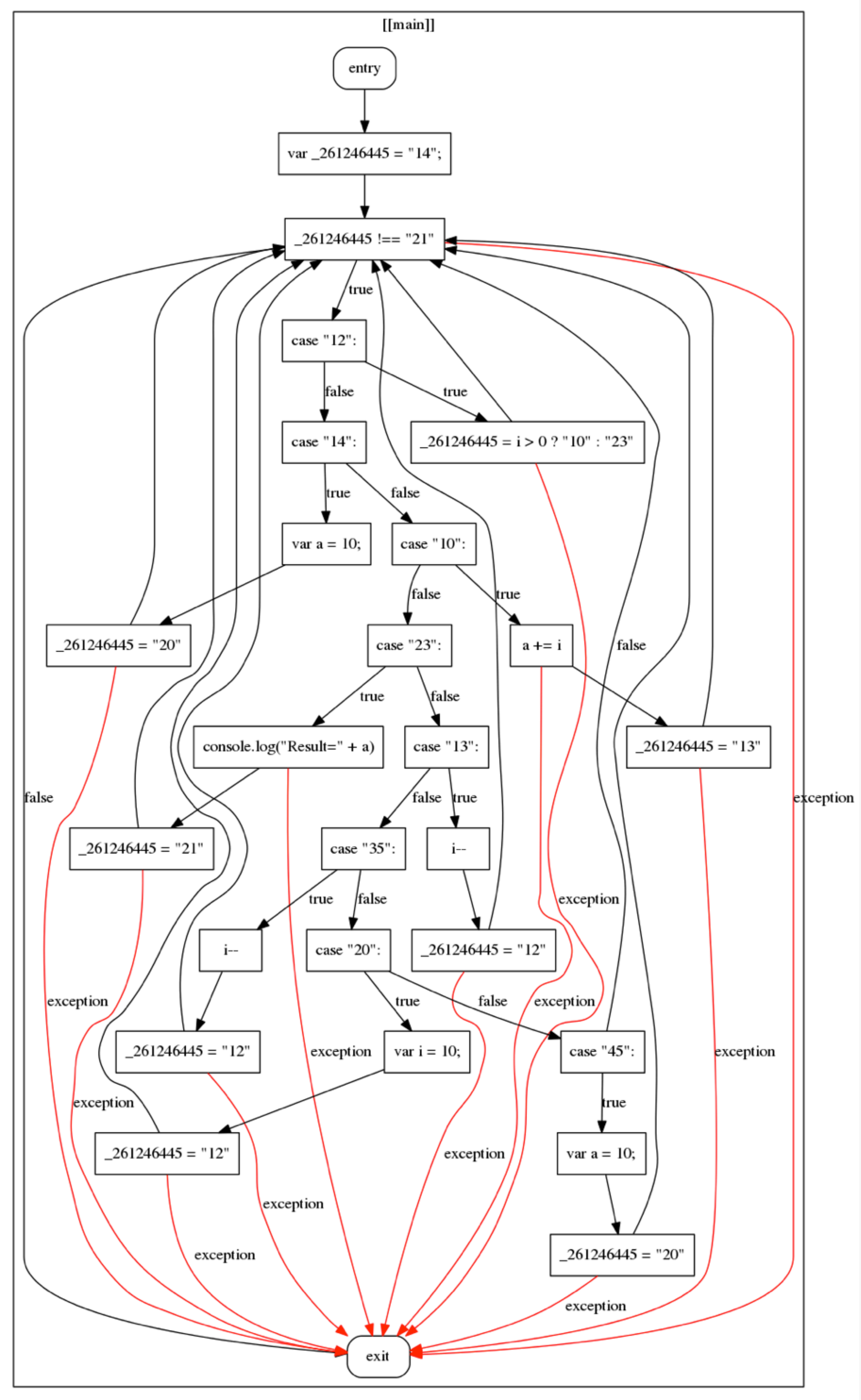




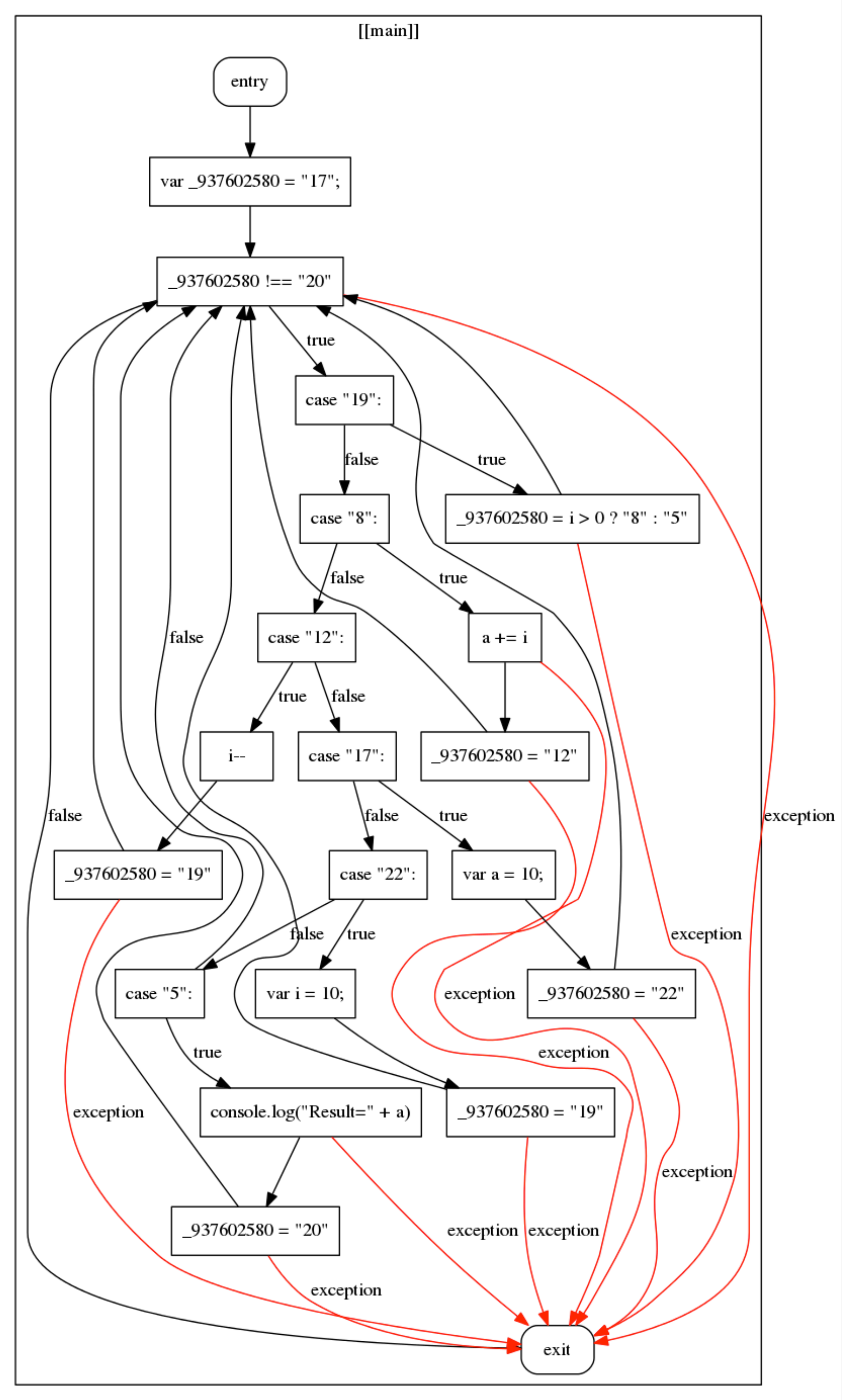
Control Flow Flattening (with Dead Clones)



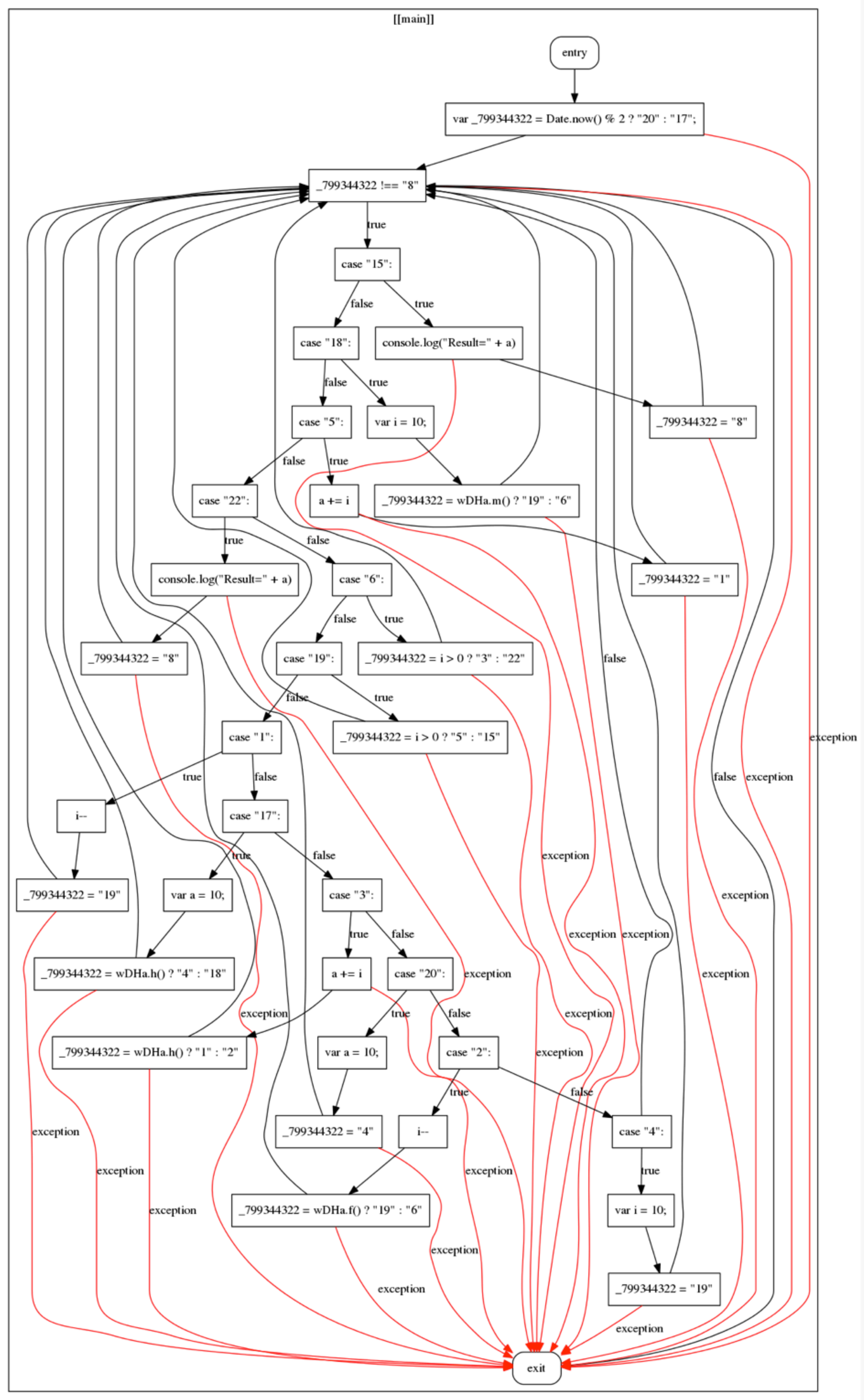
- Dead clones increase the potency
- Cheap
- File Size Increase



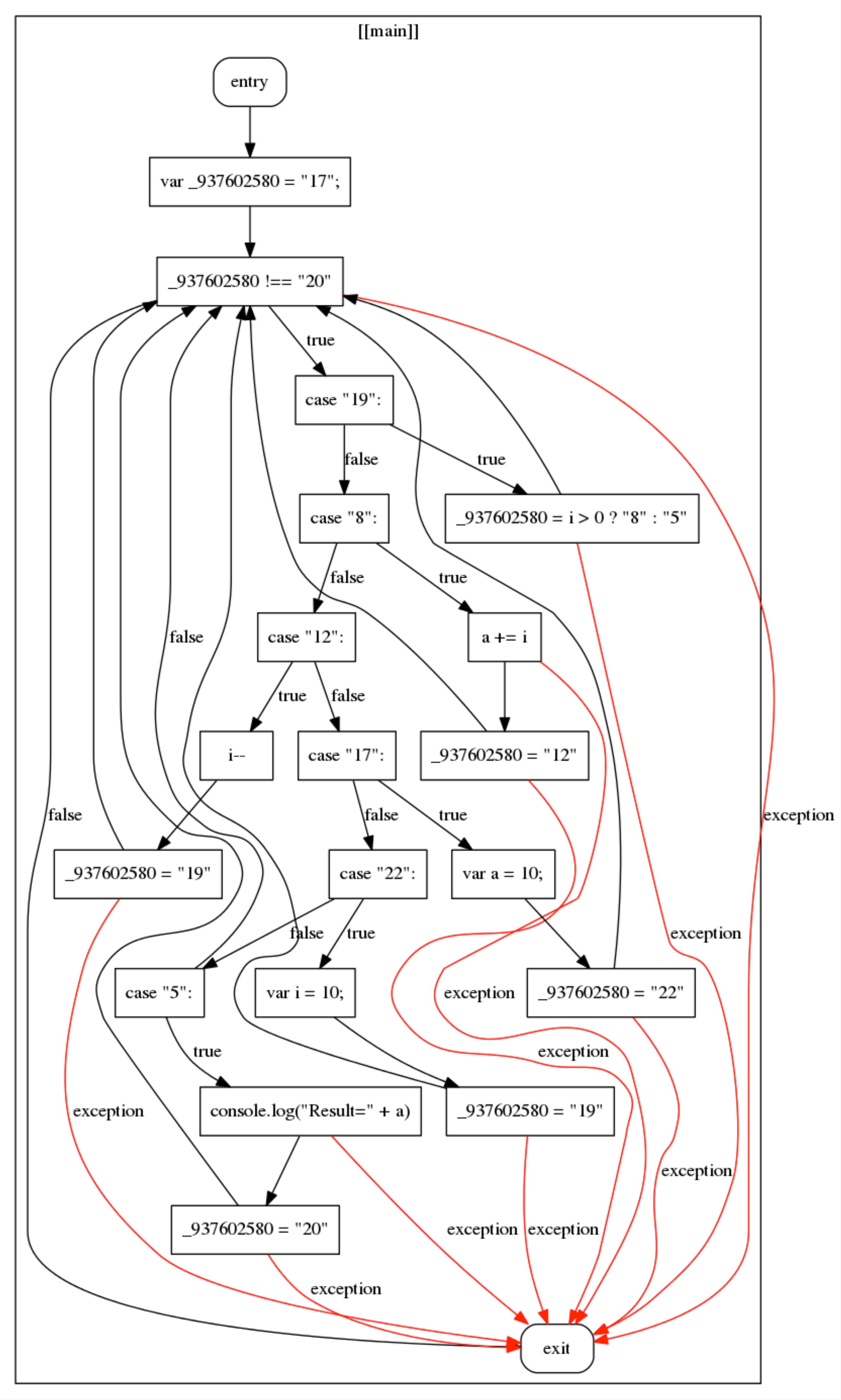
Control Flow Flattening (with Clones)



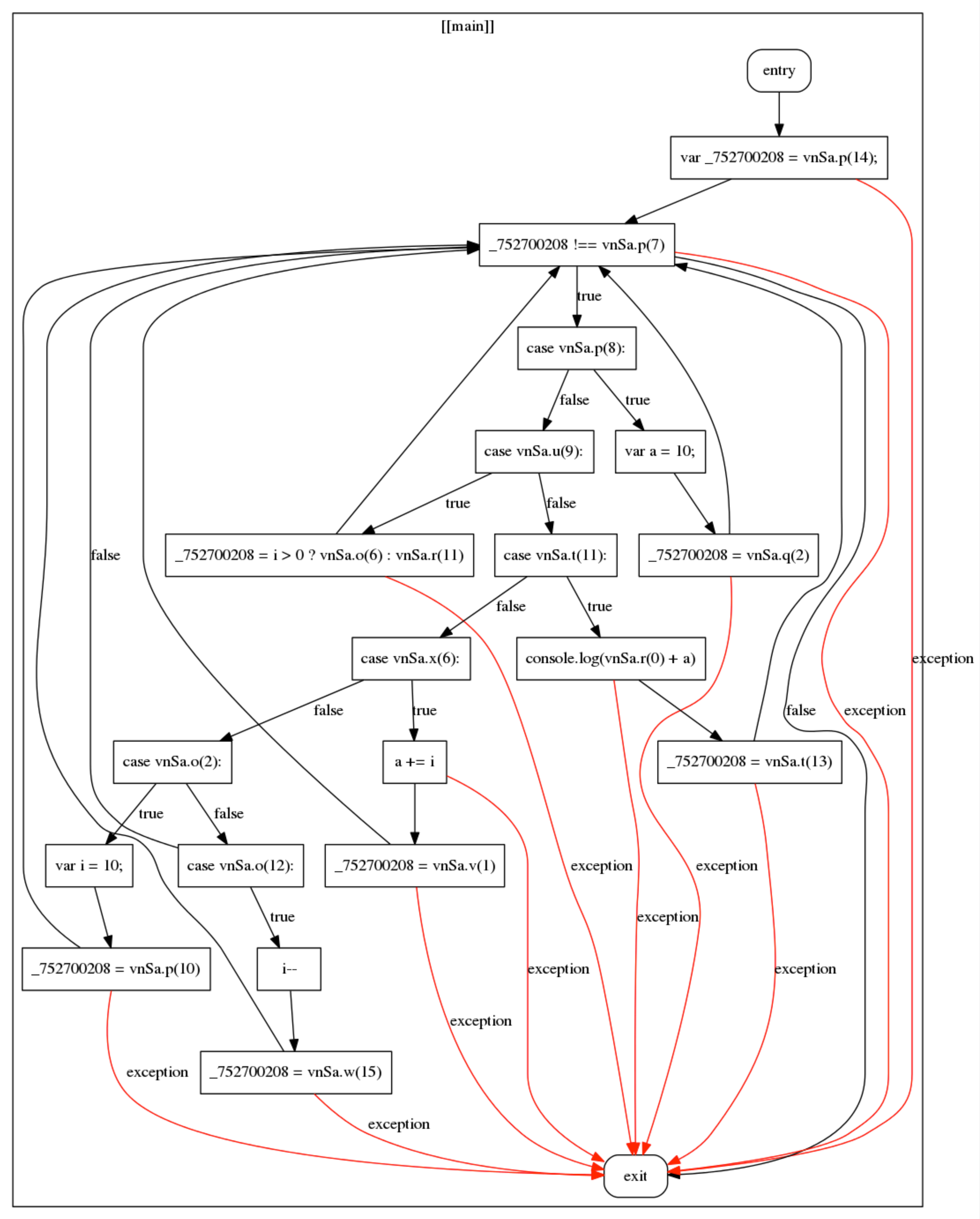
- (real) Clones increase the potency and resilience



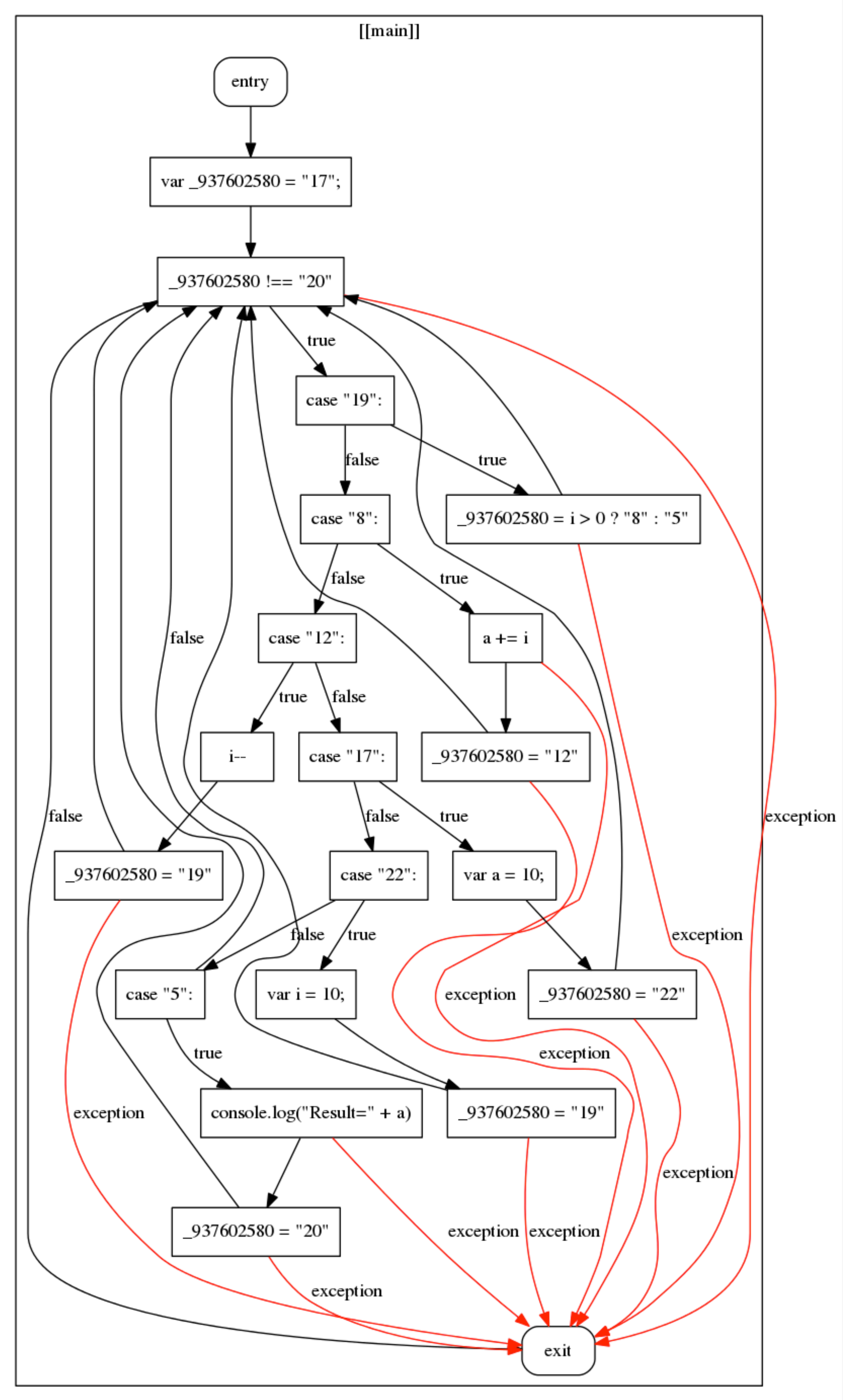
Control Flow Flattening (with Opaque Predicates)



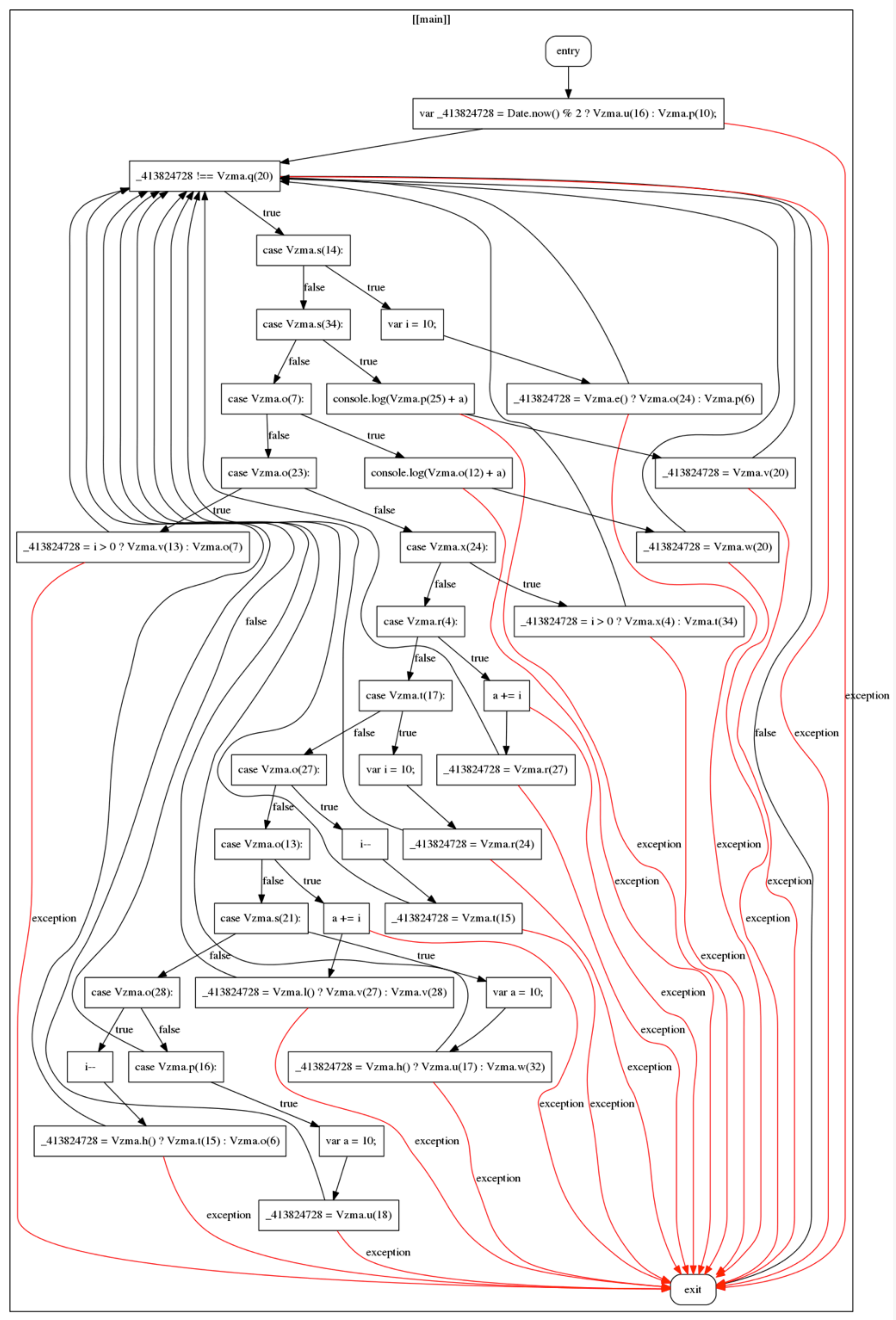
- Improved resilience



Control Flow Flattening (all options)



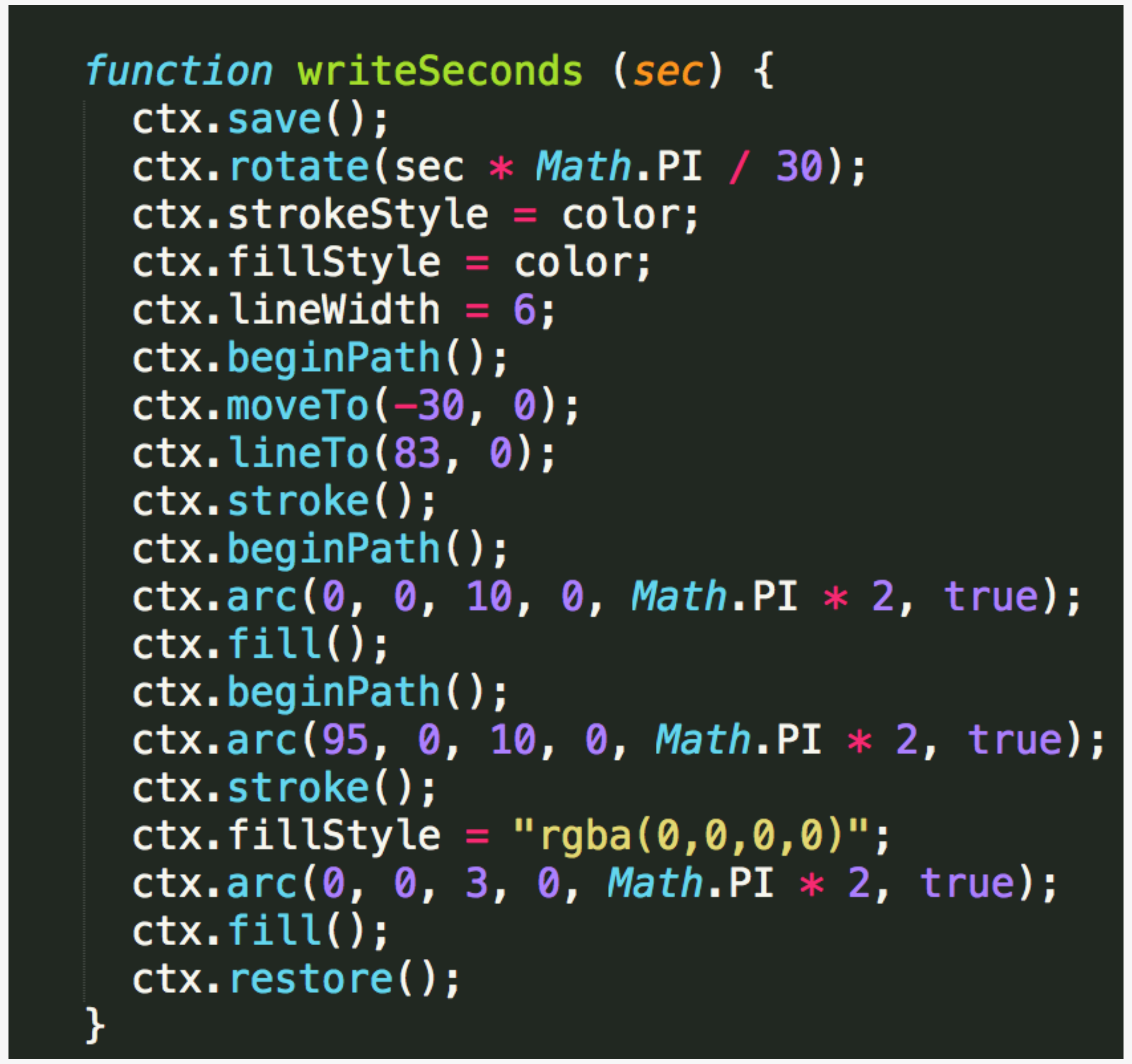
- Maximized resilience
- Even better if polymorphic



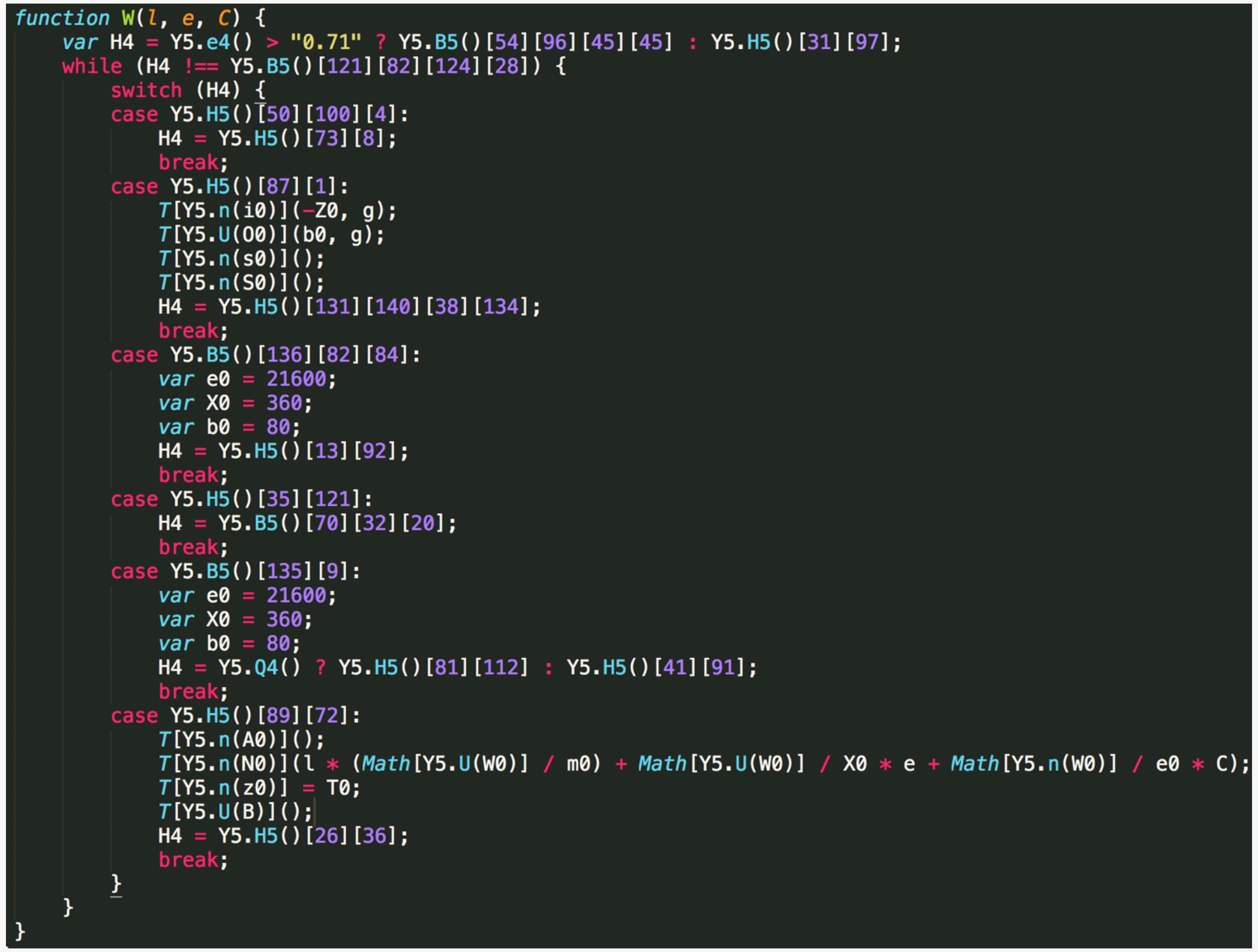
Transformation Example #5

Dot to bracket notation + Duplicate Literals Removal + String Splitting & Concealing + Identifiers Renaming + Control Flow Flattening + Function Reordering + Function Outlining

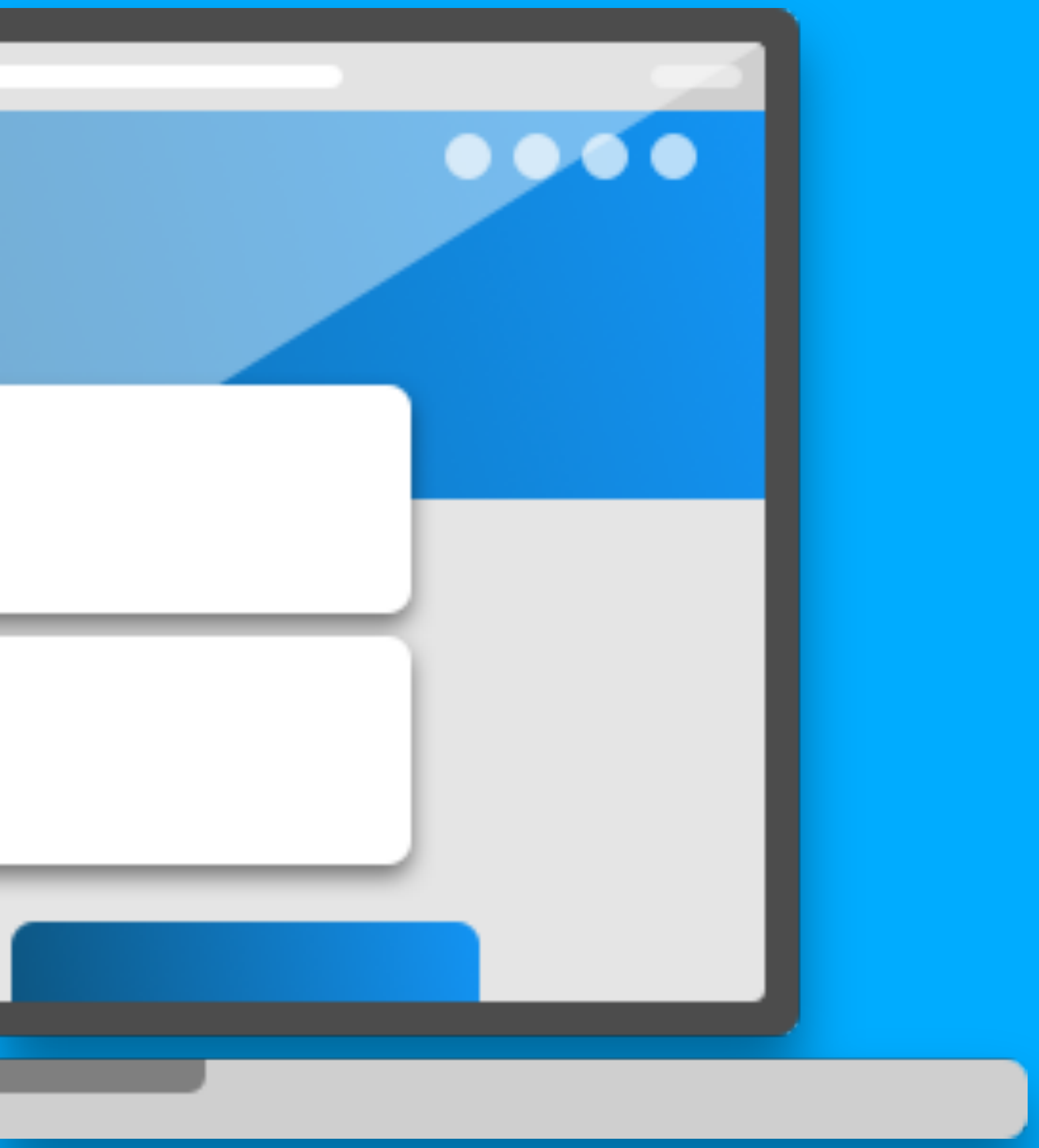
- Eliminated strings and objects names
- But we haven't really changed the control flow that much



```javascript
function writeSeconds (sec) {
    ctx.save();
    ctx.rotate(sec * Math.PI / 30);
    ctx.strokeStyle = color;
    ctx.fillStyle = color;
    ctx.lineWidth = 6;
    ctx.beginPath();
    ctx.moveTo(-30, 0);
    ctx.lineTo(83, 0);
    ctx.stroke();
    ctx.beginPath();
    ctx.arc(0, 0, 10, 0, Math.PI * 2, true);
    ctx.fill();
    ctx.beginPath();
    ctx.arc(95, 0, 10, 0, Math.PI * 2, true);
    ctx.stroke();
    ctx.fillStyle = "rgba(0,0,0,0)";
    ctx.arc(0, 0, 3, 0, Math.PI * 2, true);
    ctx.fill();
    ctx.restore();
}
```

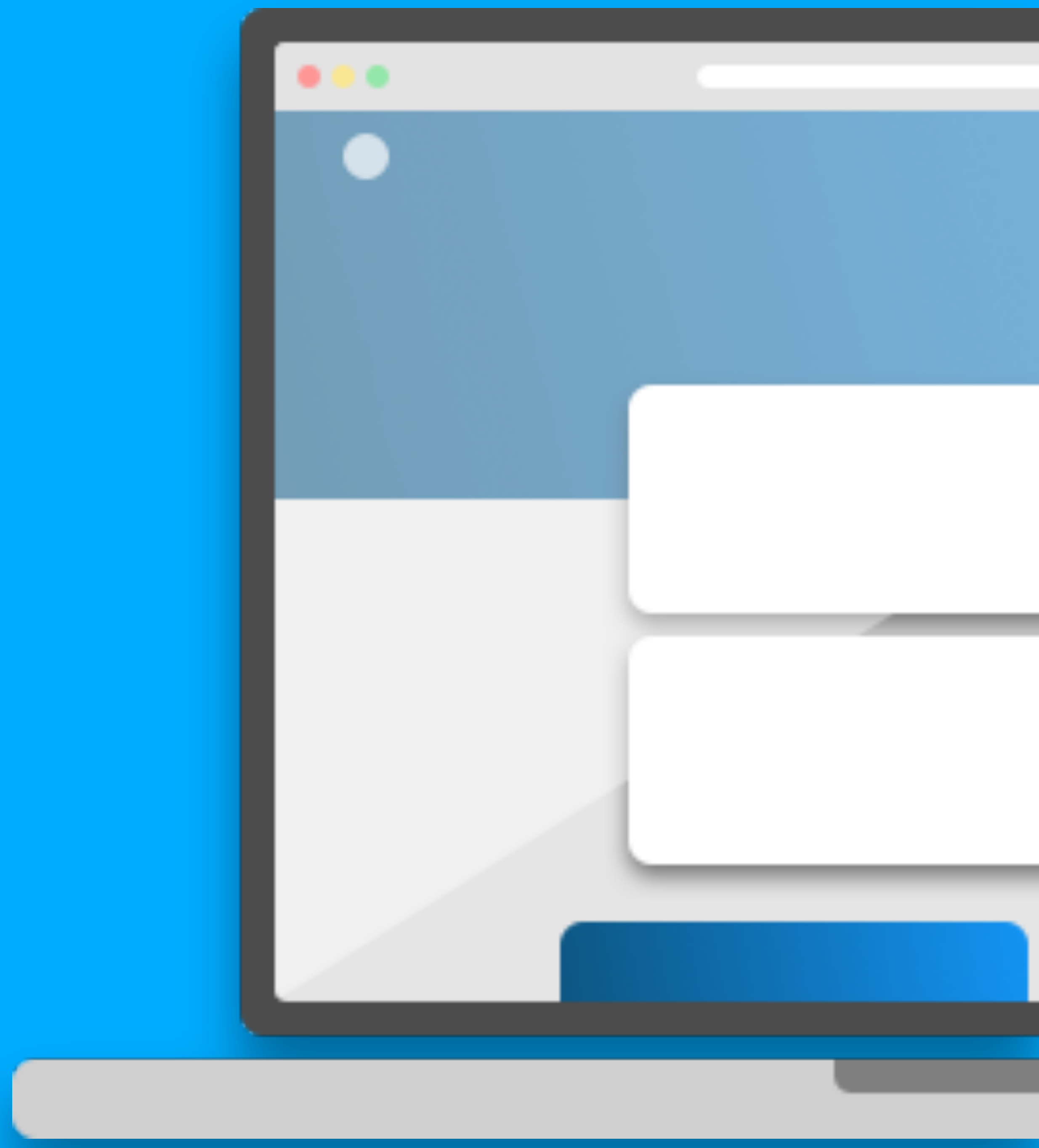


```javascript
function W(l, e, C) {
    var H4 = Y5.e4() > "0.71" ? Y5.B5()[54][96][45][45] : Y5.H5()[31][97];
    while (H4 !== Y5.B5()[121][82][124][28]) {
        switch (H4) {
        case Y5.H5()[50][100][4]:
            H4 = Y5.H5()[73][8];
            break;
        case Y5.H5()[87][1]:
            T[Y5.n(i0)](-Z0, g);
            T[Y5.U(O0)](b0, g);
            T[Y5.n(s0)]();
            T[Y5.n(S0)]();
            H4 = Y5.H5()[131][140][38][134];
            break;
        case Y5.B5()[136][82][84]:
            var e0 = 21600;
            var X0 = 360;
            var b0 = 80;
            H4 = Y5.H5()[13][92];
            break;
        case Y5.H5()[35][121]:
            H4 = Y5.B5()[70][32][20];
            break;
        case Y5.B5()[135][9]:
            var e0 = 21600;
            var X0 = 360;
            var b0 = 80;
            H4 = Y5.Q4() ? Y5.H5()[81][112] : Y5.H5()[41][91];
            break;
        case Y5.H5()[89][72]:
            T[Y5.n(A0)]();
            T[Y5.n(N0)](l * (Math[Y5.U(W0)] / m0) + Math[Y5.U(W0)] / X0 * e + Math[Y5.n(W0)] / e0 * C);
            T[Y5.n(z0)] = T0;
            T[Y5.U(B)]();
            H4 = Y5.H5()[26][36];
            break;
        }
    }
}
```



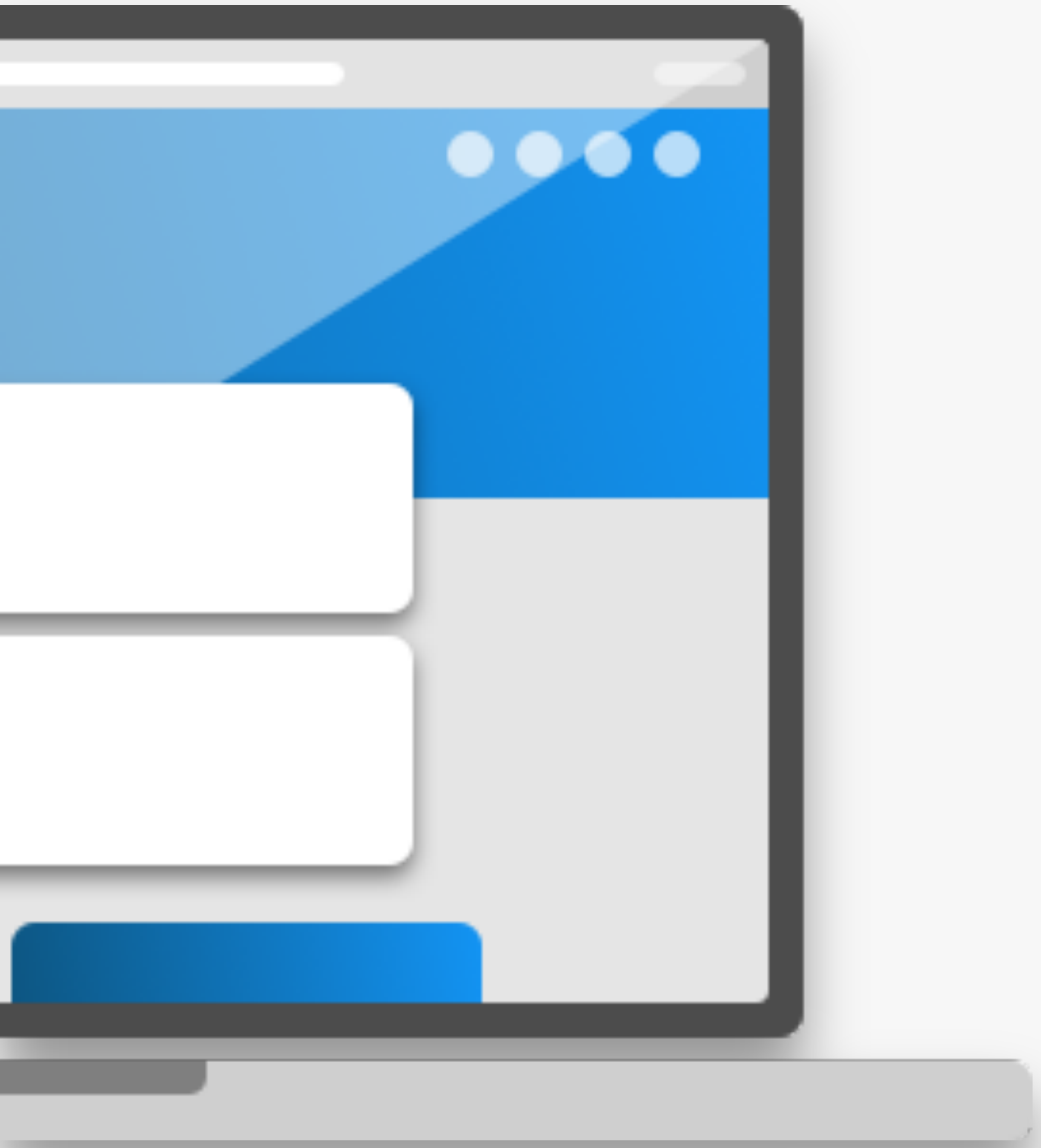
BEYOND OBFUSCATION

PART 5



jscrambler

What's Beyond



Code Traps

Tamper-resistant

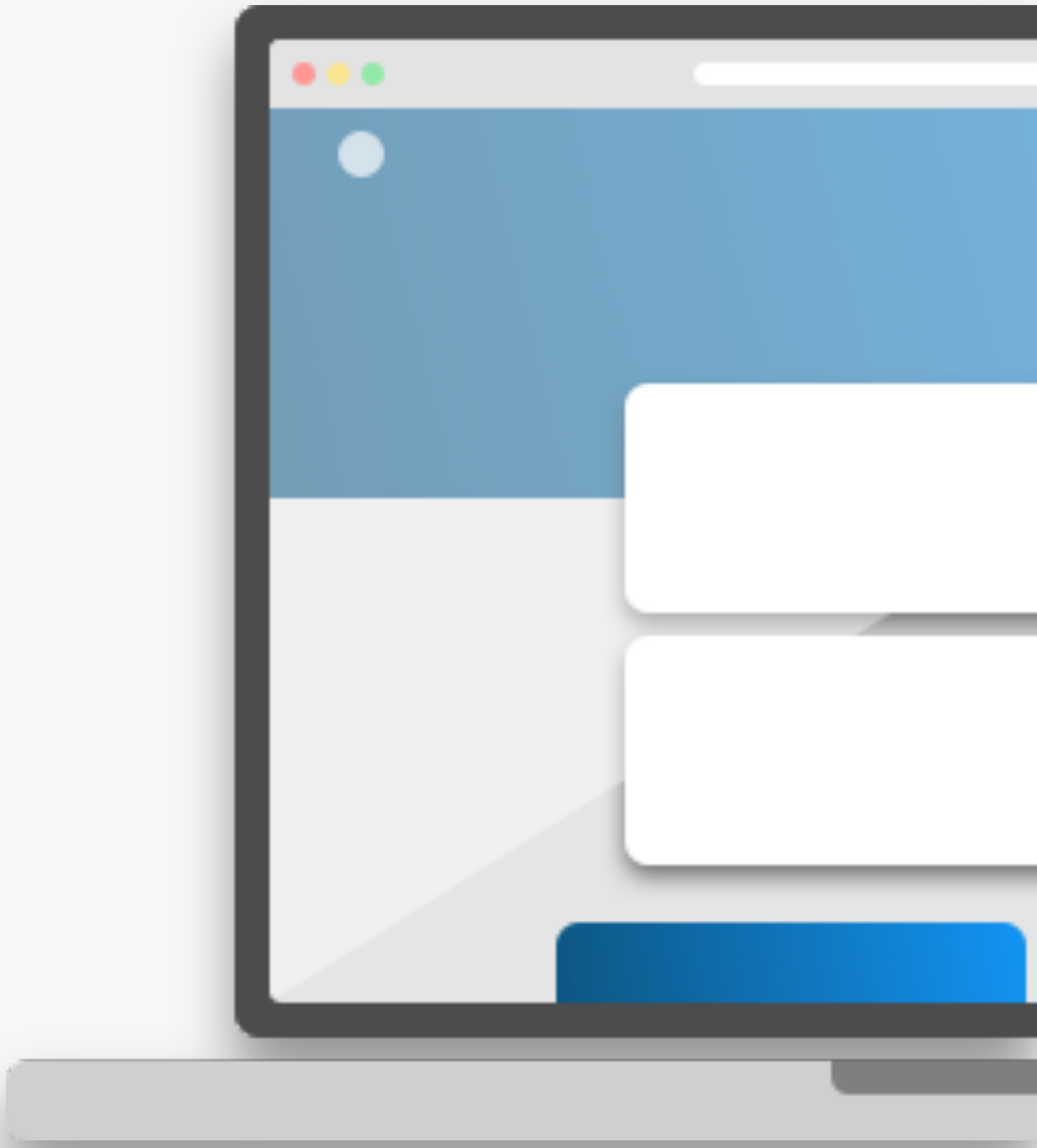
Anti-debugging

Anti-poisoning

Anti-emulation

Self-healing

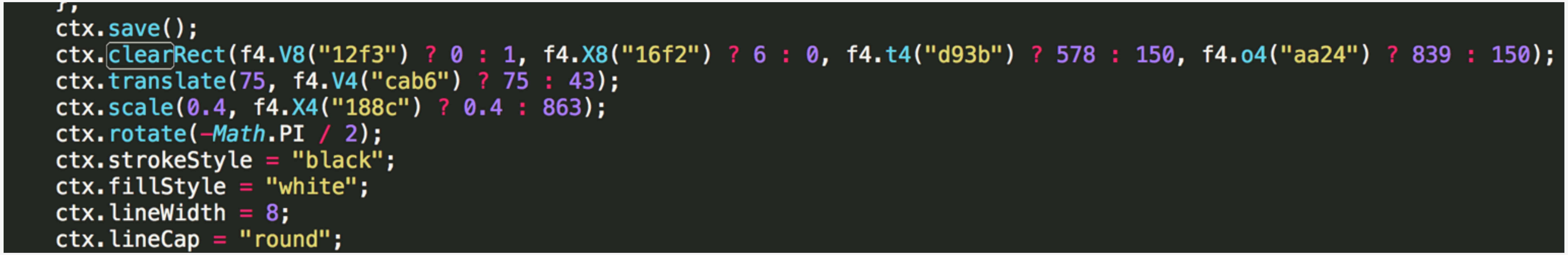
...



Code Traps

- **Added logic to enforce a certain restriction**
- **Scattered**
- **They can work together**
- **Ideally applied to different targets**
- **Similarly looking to other constructs**
- **Upon detection, multiple reactions can occur**
    - Break
    - Derail program execution
    - Redirect, refresh
    - Delete cookies
    - Alert

- **Examples**
    - Expiration date
    - Domain lock
    - OS lock (e.g. Android)
    - Browser lock (e.g. Chrome)

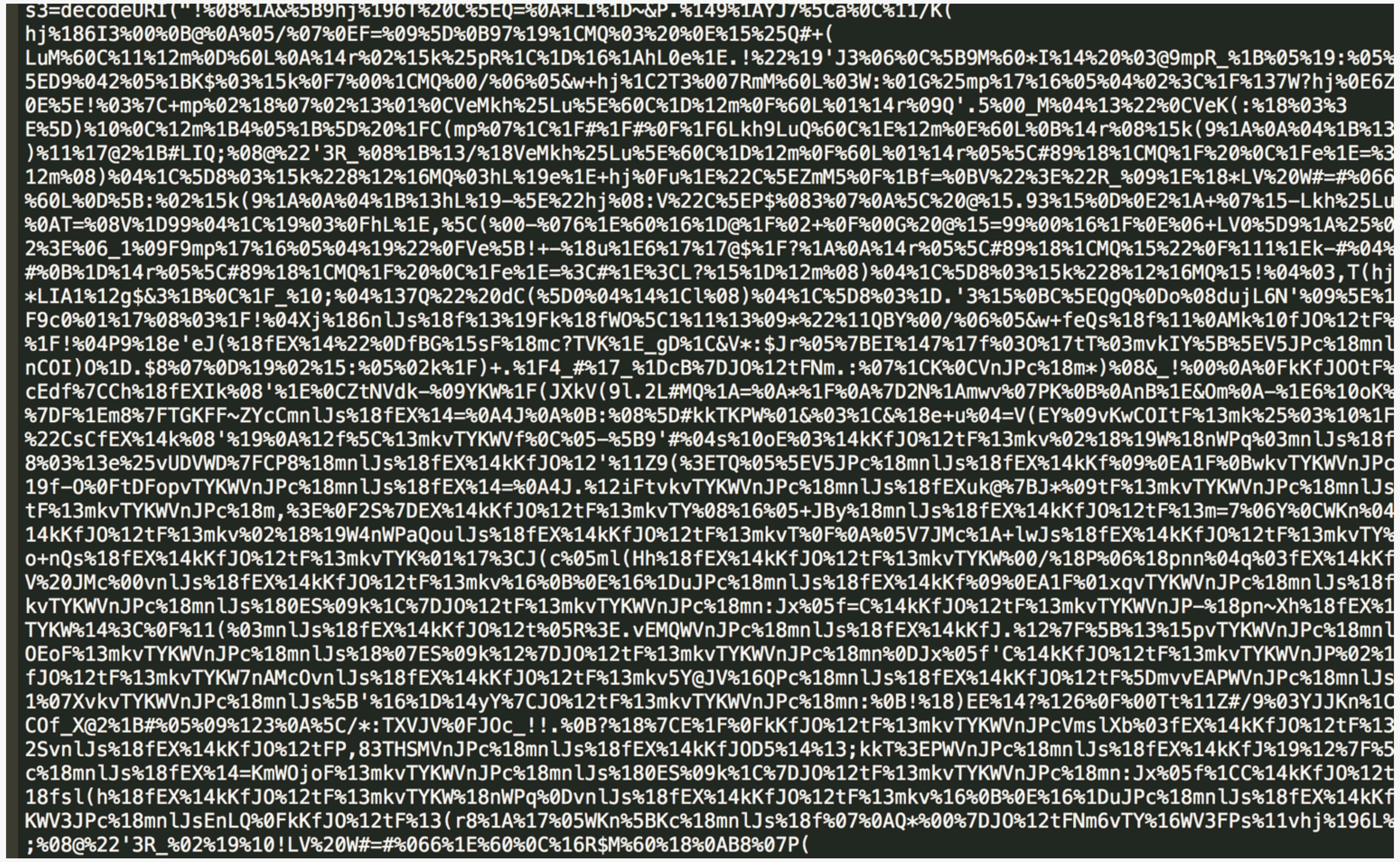


```
    };
    ctx.save();
    ctx.clearRect(f4.V8("12f3") ? 0 : 1, f4.X8("16f2") ? 6 : 0, f4.t4("d93b") ? 578 : 150, f4.o4("aa24") ? 839 : 150);
    ctx.translate(75, f4.V4("cab6") ? 75 : 43);
    ctx.scale(0.4, f4.X4("188c") ? 0.4 : 863);
    ctx.rotate(-Math.PI / 2);
    ctx.strokeStyle = "black";
    ctx.fillStyle = "white";
    ctx.lineWidth = 8;
    ctx.lineCap = "round";
```

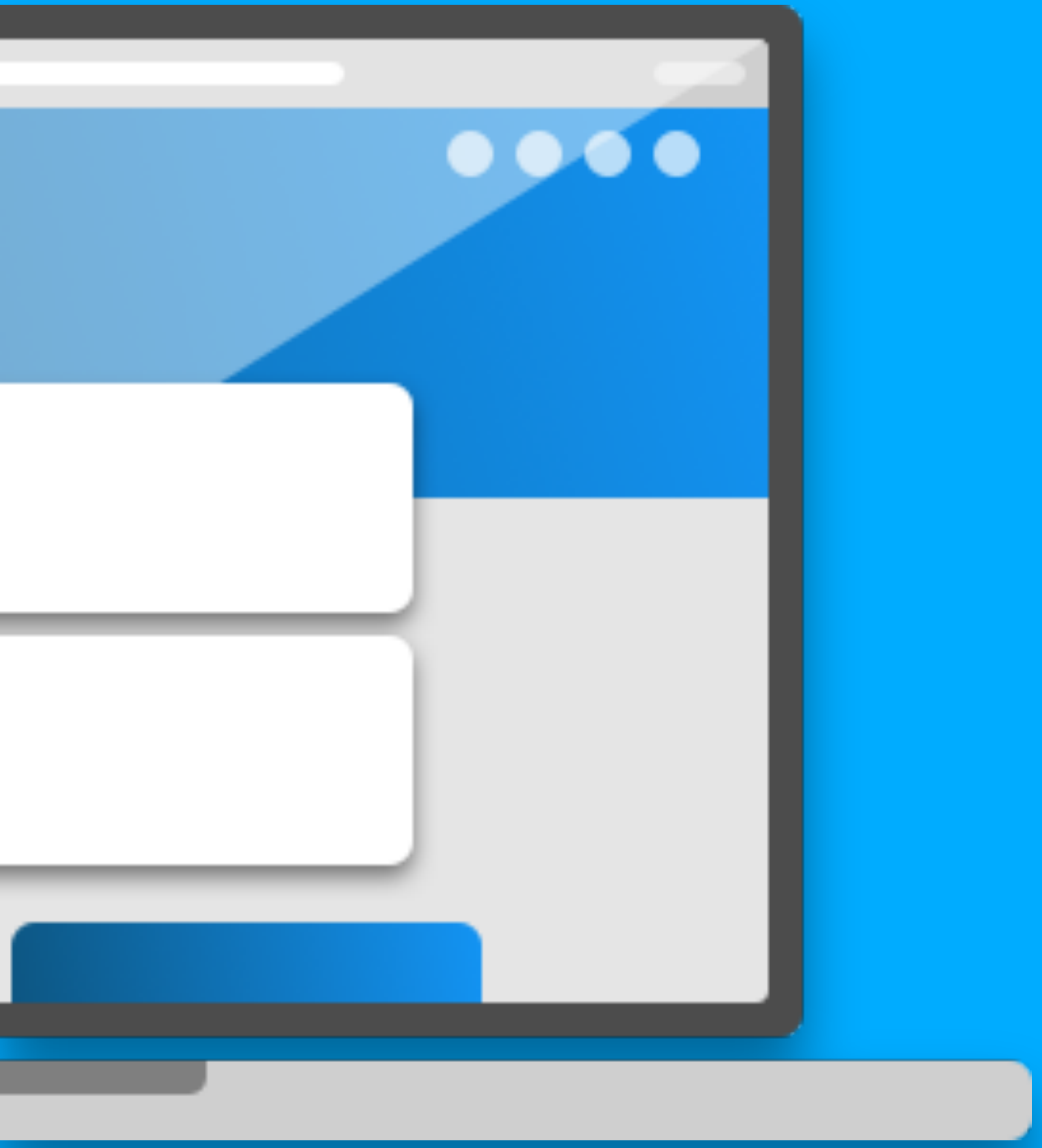
Self-defending Code

- **Anti-tampering**
    - Integrity checks
    - Can be based on checksums
    - May use introspection and embedded checksums
    - <or> Remote attestation
    - Upon detection breaks the code
    - Usually combined with other active defense techniques such as anti-debugging

- **Self-defending**
    - Aims to detect debugger use
    - Can be time-based
    - Can look for hints that the debugger is being used

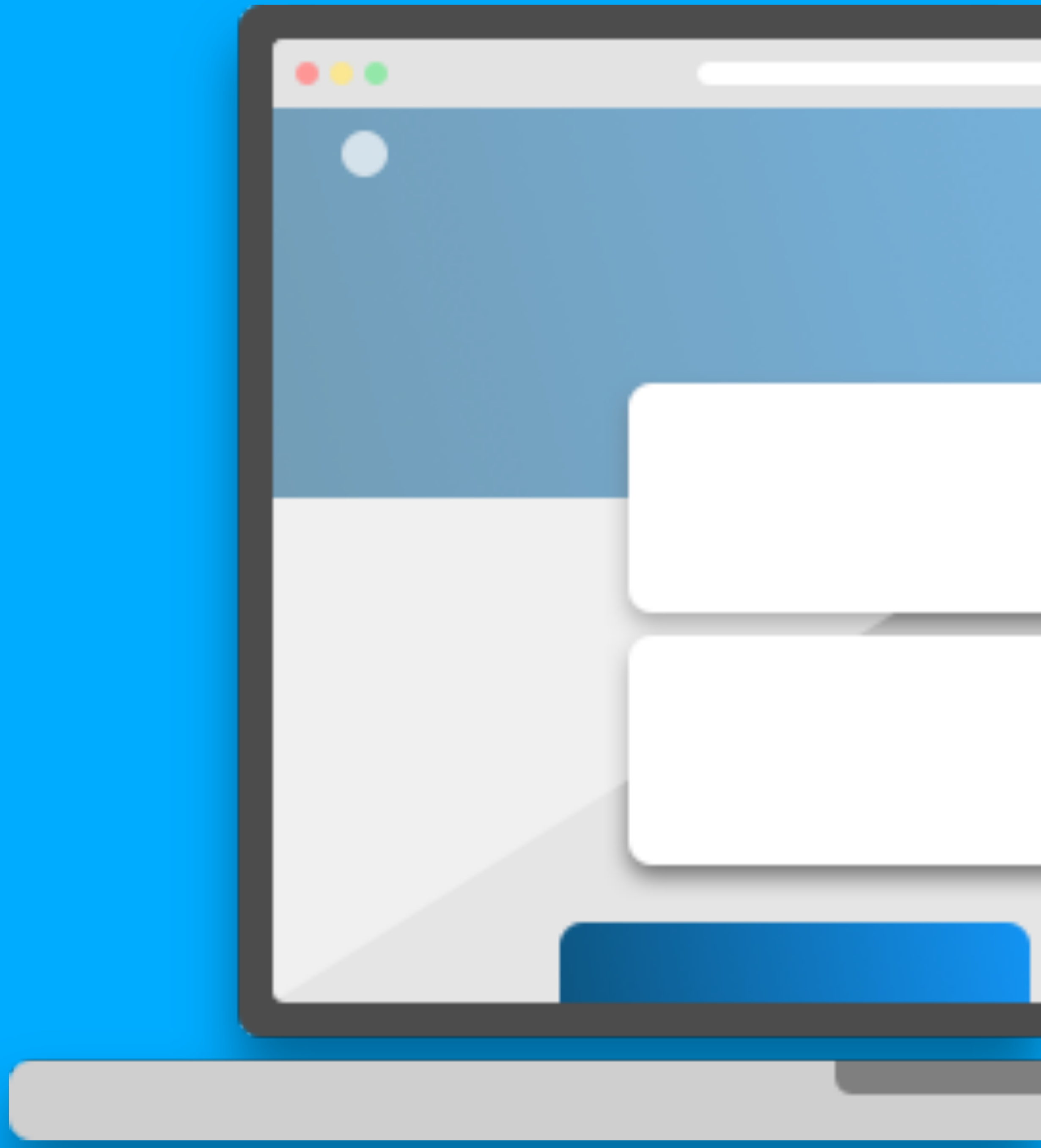


```
s3=decodeURI("!%08%1A&%5B9hj%196T%20C%5EQ=%0A*LI%1D~&P.%149%1AYJ7%5Ca%0C%11/K(
hj%186I3%00%0B@%0A%05/%07%0EF=%09%5D%0B97%19%1CMQ%03%20%0E%15%25Q#+(
LuM%60C%11%12m%0D%60L%0A%14r%02%15k%25pR%1C%1D%16%1AhL0e%1E.!%22%19'J3%06%0C%5B9M%60*I%14%20%03@9mpR_%1B%05%19:%05%
5ED9%042%05%1BK$%03%15k%0F7%00%1CMQ%00/%06%05&w+hj%1C2T3%007RmM%60L%03W:%01G%25mp%17%16%05%04%02%3C%1F%137W?hj%0E6Z
0E%5E!%03%7C+mp%02%18%07%02%13%01%0CVeMkh%25Lu%5E%60C%1D%12m%0F%60L%01%14r%09Q'.5%00_M%04%13%22%0CVeK(:%18%03%3
E%5D)%10%0C%12m%1B4%05%1B%5D%20%1FC(mp%07%1C%1F#%1F#%0F%1F6Lkh9LuQ%60C%1E%12m%0E%60L%0B%14r%08%15k(9%1A%0A%04%1B%13
)%11%17@2%1B#LIQ;%08@%22'3R_%08%1B%13/%18VeMkh%25Lu%5E%60C%1D%12m%0F%60L%01%14r%05%5C#89%18%1CMQ%1F%20%0C%1Fe%1E=%3
12m%08)%04%1C%5D8%03%15k%228%12%16MQ%03hL%19e%1E+hj%0Fu%1E%22C%5EZmM5%0F%1Bf=%0BV%22%3E%22R_%09%1E%18*LV%20W#=#%066
%60L%0D%5B:%02%15k(9%1A%0A%04%1B%13hL%19-%5E%22hj%08:V%22C%5EP$%083%07%0A%5C%20@%15.93%15%0D%0E2%1A+%07%15-Lkh%25Lu
%0AT=%08V%1D99%04%1C%19%03%0FhL%1E,%5C(%00-%076%1E%60%16%1D@%1F%02+%0F%00G%20@%15=99%00%16%1F%0E%06+LV0%5D9%1A%25%0
2%3E%06_1%09F9mp%17%16%05%04%19%22%0FVe%5B!+-%18u%1E6%17%17@$%1F?%1A%0A%14r%05%5C#89%18%1CMQ%15%22%0F%111%1Ek-#%04%
#%0B%1D%14r%05%5C#89%18%1CMQ%1F%20%0C%1Fe%1E=%3C#%1E%3CL?%15%1D%12m%08)%04%1C%5D8%03%15k%228%12%16MQ%15!%04%03,T(hj
*LIA1%12g$&3%1B%0C%1F_%10;%04%137Q%22%20dC(%5D0%04%14%1Cl%08)%04%1C%5D8%03%1D.'3%15%0BC%5EQgQ%0Do%08dujL6N'%09%5E%1
F9c0%01%17%08%03%1F!%04Xj%186nlJs%18f%13%19Fk%18fWO%5C1%11%13%09*%22%11QBY%00/%06%05&w+feQs%18f%11%0AMk%10fJO%12tF%
%1F!%04P9%18e'eJ(%18fEX%14%22%0DfBG%15sF%18mc?TVK%1E_gD%1C&V*:$Jr%05%7BEI%147%17f%030%17tT%03mvkIY%5B%5EV5JPc%18mnl
nCOI)O%1D.$8%07%0D%19%02%15:%05%02k%1F)+.%1F4_#%17_%1DcB%7DJO%12tFNm.:%07%1CK%0CVnJPc%18m*)%08&_!%00%0A%0FkKfJOOtF%
cEdf%7CCh%18fEXIk%08'%1E%0CZtNVdk-%09YKW%1F(JXkV(9l.2L#MQ%1A=%0A*%1F%0A%7D2N%1Amwv%07PK%0B%0AnB%1E&Om%0A-%1E6%10oK%
%7DF%1Em8%7FTGKFF~ZYcCmnlJs%18fEX%14=%0A4J%0A%0B:%08%5D#kkTKPW%01&%03%1C&%18e+u%04=V(EY%09vKwCOItF%13mk%25%03%10%1F
%22CsCfEX%14k%08'%19%0A%12f%5C%13mkvTYKWVf%0C%05-%5B9'#%04s%10oE%03%14kKfJO%12tF%13mkv%02%18%19W%18nWPq%03mnlJs%18f
8%03%13e%25vUDVWD%7FCP8%18mnlJs%18fEX%14kKfJO%12'%11Z9(%3ETQ%05%5EV5JPc%18mnlJs%18fEX%14kKf%09%0EA1F%0BwkvTYKWVnJPc
19f-O%0FtDFopvTYKWVnJPc%18mnlJs%18fEX%14=%0A4J.%12iFtvkvTYKWVnJPc%18mnlJs%18fEXuk@%7BJ*%09tF%13mkvTYKWVnJPc%18mnlJs
tF%13mkvTYKWVnJPc%18m,%3E%0F2S%7DEX%14kKfJO%12tF%13mkvTY%08%16%05+JBy%18mnlJs%18fEX%14kKfJO%12tF%13m=7%06Y%0CWKn%04
14kKfJO%12tF%13mkv%02%18%19W4nWPaQoulJs%18fEX%14kKfJO%12tF%13mkvT%0F%0A%05V7JMc%1A+lwJs%18fEX%14kKfJO%12tF%13mkvTY%
o+nQs%18fEX%14kKfJO%12tF%13mkvTYK%01%17%3CJ(c%05ml(Hh%18fEX%14kKfJO%12tF%13mkvTYKW%00/%18P%06%18pnn%04q%03fEX%14kKf
V%20JMc%00vnlJs%18fEX%14kKfJO%12tF%13mkv%16%0B%0E%16%1DuJPc%18mnlJs%18fEX%14kKf%09%0EA1F%01xqvTYKWVnJPc%18mnlJs%18f
kvTYKWVnJPc%18mnlJs%180ES%09k%1C%7DJO%12tF%13mkvTYKWVnJPc%18mn:Jx%05f=C%14kKfJO%12tF%13mkvTYKWVnJP-%18pn~Xh%18fEX%1
TYKW%14%3C%0F%11(%03mnlJs%18fEX%14kKfJO%12t%05R%3E.vEMQWVnJPc%18mnlJs%18fEX%14kKfJ.%12%7F%5B%13%15pvTYKWVnJPc%18mnl
OEoF%13mkvTYKWVnJPc%18mnlJs%18%07ES%09k%12%7DJO%12tF%13mkvTYKWVnJPc%18mn%0DJx%05f'C%14kKfJO%12tF%13mkvTYKWVnJP%02%1
fJO%12tF%13mkvTYKW7nAMcOvnlJs%18fEX%14kKfJO%12tF%13mkv5Y@JV%16QPc%18mnlJs%18fEX%14kKfJO%12tF%5DmvvEAPWVnJPc%18mnlJs
1%07XvkvTYKWVnJPc%18mnlJs%5B'%16%1D%14yY%7CJO%12tF%13mkvTYKWVnJPc%18mn:%0B!%18)EE%14?%126%0F%00Tt%11Z#/9%03YJJKn%1C
COf_X@2%1B#%05%09%123%0A%5C/*:TXVJV%0FJOc_!!.%0B?%18%7CE%1F%0FkKfJO%12tF%13mkvTYKWVnJPcVmslXb%03fEX%14kKfJO%12tF%13
2SvnlJs%18fEX%14kKfJO%12tFP,83THSMVnJPc%18mnlJs%18fEX%14kKfJOD5%14%13;kkT%3EPWVnJPc%18mnlJs%18fEX%14kKfJ%19%12%7F%5
c%18mnlJs%18fEX%14=KmWOjoF%13mkvTYKWVnJPc%18mnlJs%180ES%09k%1C%7DJO%12tF%13mkvTYKWVnJPc%18mn:Jx%05f%1CC%14kKfJO%12t
18fsl(h%18fEX%14kKfJO%12tF%13mkvTYKW%18nWPq%0DvnlJs%18fEX%14kKfJO%12tF%13mkv%16%0B%0E%16%1DuJPc%18mnlJs%18fEX%14kKf
KWV3JPc%18mnlJsEnLQ%0FkKfJO%12tF%13(r8%1A%17%05WKn%5BKc%18mnlJs%18f%07%0AQ*%00%7DJO%12tFNm6vTY%16WV3FPs%11vhj%196L%
;%08@%22'3R_%02%19%10!LV%20W#=#%066%1E%60%0C%16R$M%60%18%0AB8%07P(
```



CONCLUSIONS

PART 6



jscrambler

Conclusions

Apart from legal, the only solution to protect against Reverse Engineering when physical access is given to the software (MATE attacks)

e.g.'s Mobile applications, on prem, desktop, etc

Obfuscation value depends on

The sophistication of the code transformations

The power of the available deobfuscation techniques

The amount of resources available (time, motivation, money, etc) to the attacker

Conclusions Continued

Obfuscation potency is important, but resilience is more

But people often evaluate obfuscation merely based on its potency (not real)

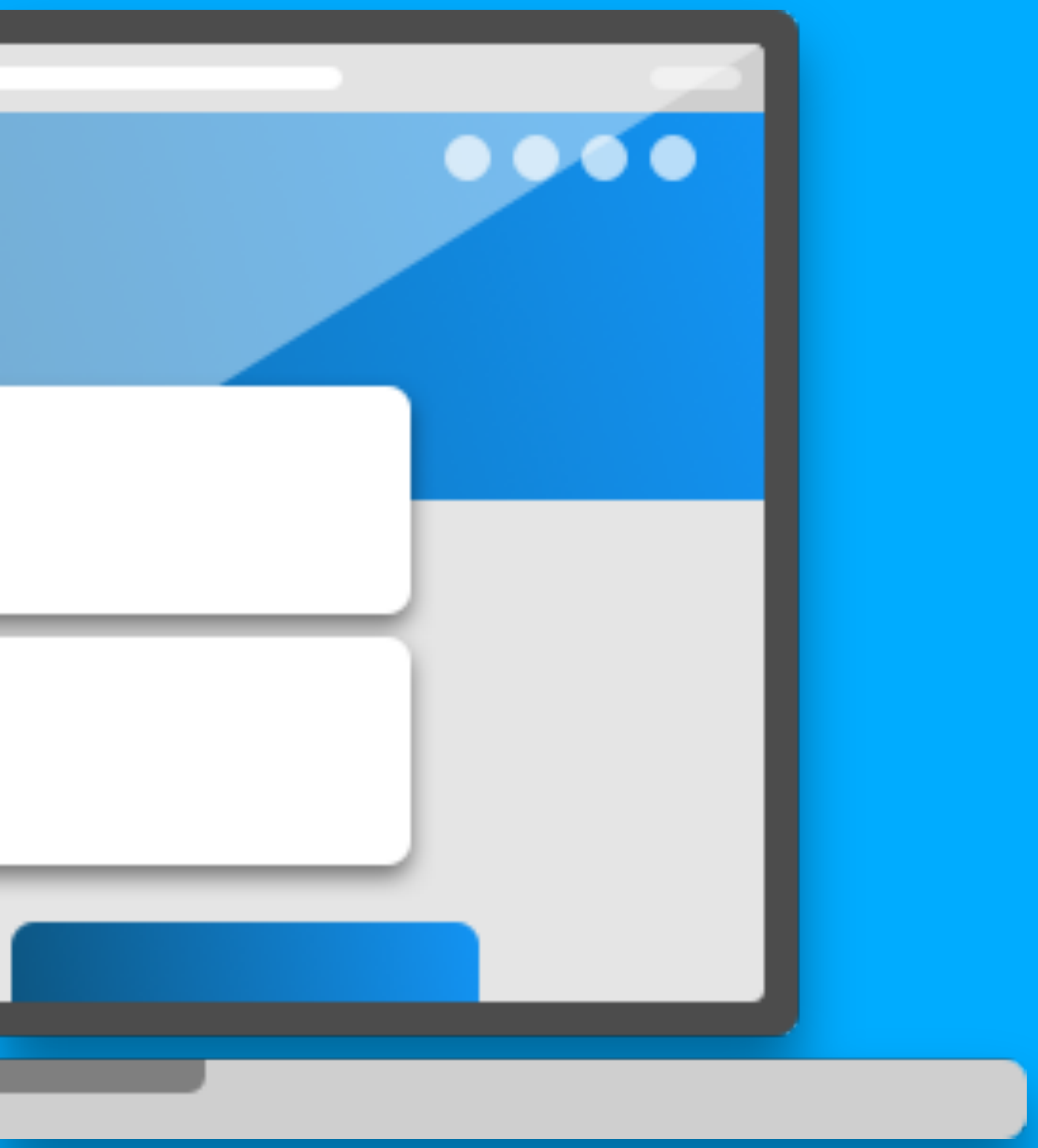
Evaluating resilience is hard (check session #2)

Control FlowObfuscation combined with strong resilient Opaque predicates is essential

Diversity is important => can help preclude attack automation

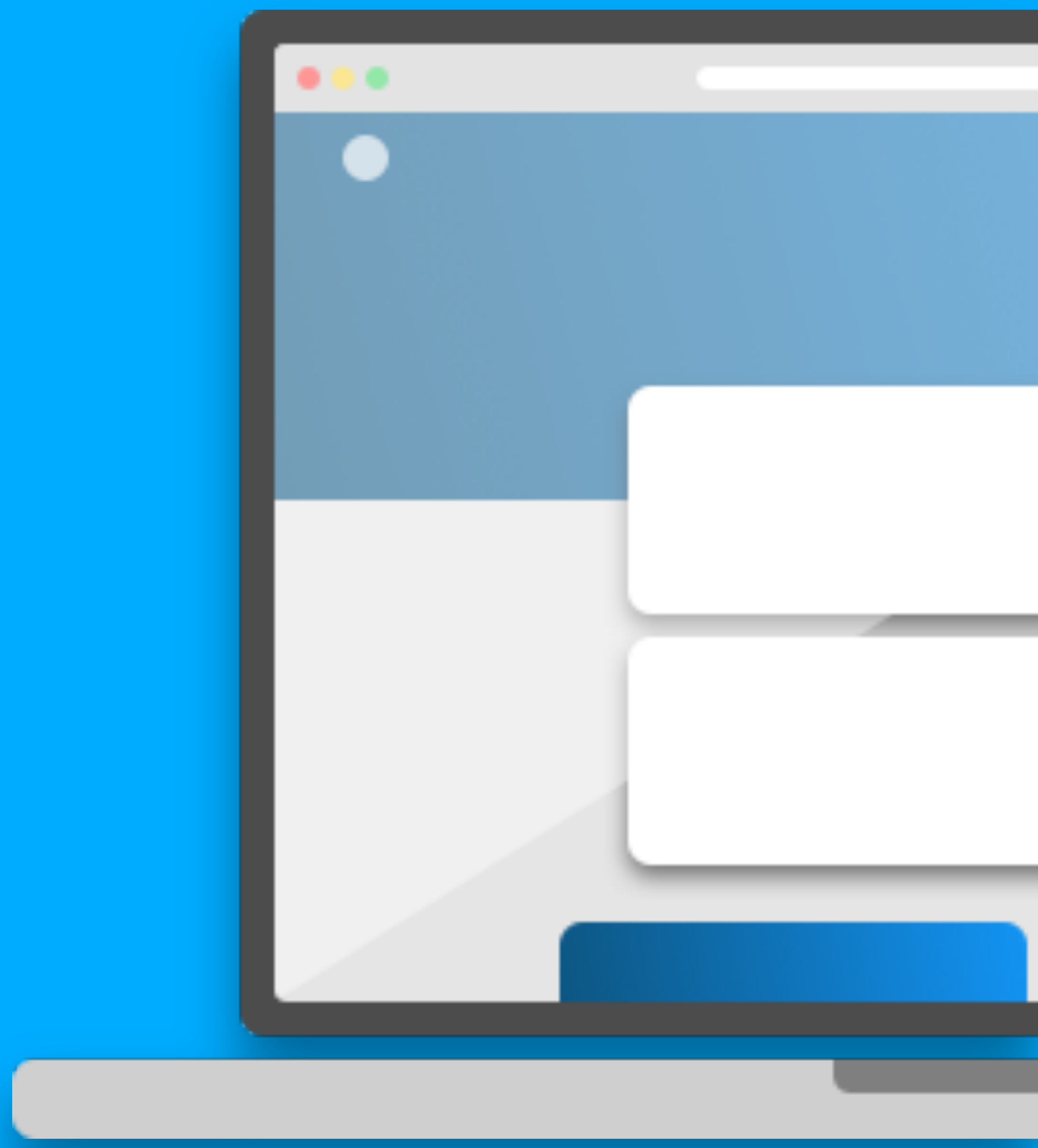
Success in using obfuscation requires searching for good tradeoffs for specific applications

Tamper-resistant code takes code protection resilience to the next level



THANK YOU!

@pedrofortuna



jscrambler