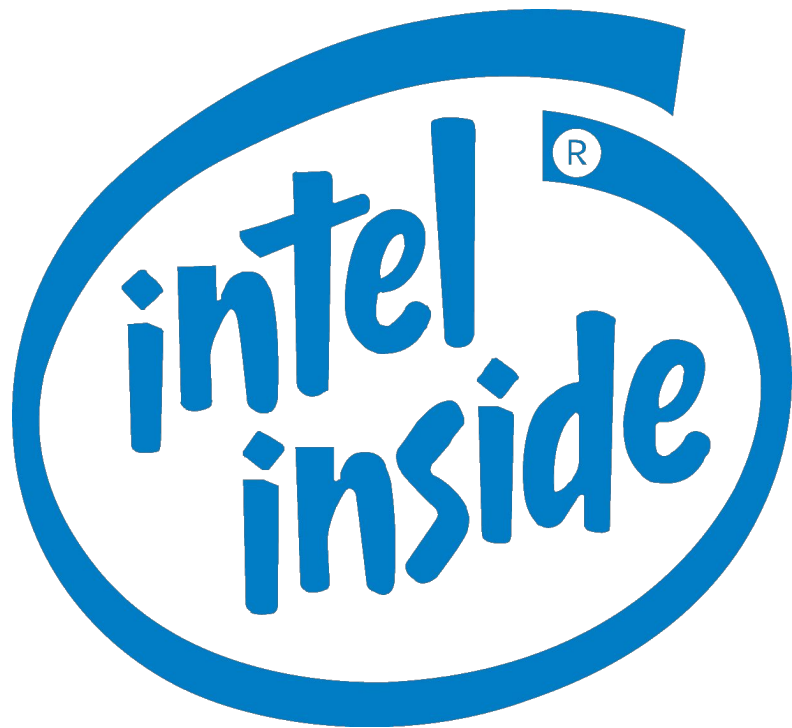


Programming Languages



machine code

01111111	01000101	01001100	01000110	00000010	00000001	00000001	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000001	00000000	00111110	00000000	00000001	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
00101000	00000010	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	01000000	00000000	00000000	00000000
00000000	00000000	01000000	00000000	00001010	00000000	00000001	00000000
01010101	01001000	10001001	11100101	01001000	10000011	11101100	00010000
01001000	10111111	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	10110000	00000000	11101000	00000000	00000000	00000000
00000000	00110001	11001001	10001001	01000101	11111100	10001001	11001000
01001000	10000011	11000100	00010000	01011101	11000011	01101000	01100101
01101100	01101100	01101111	00101100	00100000	01110111	01101111	01110010
01101100	01100100	00001010	00000000	00000000	01100011	01101100	01100001
01101110	01100111	00100000	01110110	01100101	01110010	01110011	01101001
...							

source code

```

...
main:                                     # @main
    .cfi_startproc
# %bb.0:
    pushq    %rbp
    .cfi_def_cfa_offset 16
    .cfi_offset %rbp, -16
    movq     %rsp, %rbp
    .cfi_def_cfa_register %rbp
    subq     $16, %rsp
    movabsq  $.L.str, %rdi
    movb     $0, %al
    callq    printf
    xorl     %ecx, %ecx
    movl     %eax, -4(%rbp)              # 4-byte Spill
    movl     %ecx, %eax
    addq     $16, %rsp
    popq     %rbp
    retq
.Lfunc_end0:
    .size    main, .Lfunc_end0-main
    .cfi_endproc
                                     # -- End function
    .type    .L.str,@object           # @.str
    .section      .rodata.str1.1,"aMS",@progbits,1
.L.str:
    .asciz   "hello, world\n"
    .size    .L.str, 14
...

```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    printf("hello, world\n");
```

```
}
```

compiler

interpreter

```
print("hello, world")
```

bytecode

2	0 LOAD_GLOBAL	0 (print)
	3 LOAD_CONST	1 ('hello, world')
	6 CALL_FUNCTION	1 (1 positional, 0 keyword pair)
	9 POP_TOP	
	10 LOAD_CONST	0 (None)
	13 RETURN_VALUE	

```
class Hello
{
    public static void main(String [] args)
    {
        System.out.println("hello, world");
    }
}
```

virtual machine

```
class Hello {
    Hello();
    Code:
        0: aload_0
        1: invokespecial #1          // Method java/lang/Object."<init>":()V
        4: return

    public static void main(java.lang.String[]);
    Code:
        0: getstatic      #2          // Field java/lang/System.out:Ljava/io/PrintStream;
        3: ldc            #3          // String hello, world
        5: invokevirtual #4          // Method java/io/PrintStream.println:(Ljava/lang/String;)V
        8: return
}
```

```
console.log("hello, world")
```



```
put "hello, world"
```

```
#include <iostream>
```

```
int main()
```

```
{
```

```
    std::cout << "hello, world" << std::endl;
```

```
}
```

helloworldcollection.de

- Bash
- C
- C++
- C#
- Clojure
- Erlang
- F#
- Go
- Haskell
- Java
- JavaScript
- Objective-C
- OCaml

- PHP
- Python
- R
- Ruby
- Scala
- Scheme
- SQL
- Swift
- ...

wikipedia.org/wiki/List_of_programming_languages

```
0  pick up phone book
1  open to middle of phone book
2  look at names
3  if Smith is among names
4      call Mike
5  else if Smith is earlier in book
6      open to middle of left half of book
7      go back to step 2
8  else if Smith is later in book
9      open to middle of right half of book
10     go back to step 2
11 else
12     quit
```

```
0  pick up phone book
1  open to middle of phone book
2  look at names
3  if Smith is among names
4      call Mike
5  else if Smith is earlier in book
6      open to middle of left half of book
7      go back to step 2
8  else if Smith is later in book
9      open to middle of right half of book
10     go back to step 2
11 else
12     quit
```

```
0  pick up phone book
1  open to middle of phone book
2  look at names
3  if Smith is among names
4      call Mike
5  else if Smith is earlier in book
6      open to middle of left half of book
7      go back to step 2
8  else if Smith is later in book
9      open to middle of right half of book
10     go back to step 2
11 else
12     quit
```



```
0  pick up phone book
1  open to middle of phone book
2  look at names
3  if Smith is among names
4      call Mike
5  else if Smith is earlier in book
6      open to middle of left half of book
7      go back to step 2
8  else if Smith is later in book
9      open to middle of right half of book
10     go back to step 2
11 else
12     quit
```

```
0  pick up phone book
1  open to middle of phone book
2  look at names
3  if Smith is among names
4      call Mike
5  else if Smith is earlier in book
6      open to middle of left half of book
7      go back to step 2
8  else if Smith is later in book
9      open to middle of right half of book
10     go back to step 2
11 else
12     quit
```

```
0  pick up phone book
1  open to middle of phone book
2  look at names
3  if Smith is among names
4      call Mike
5  else if Smith is earlier in book
6      open to middle of left half of book
7      go back to step 2
8  else if Smith is later in book
9      open to middle of right half of book
10     go back to step 2
11 else
12     quit
```

bool

float

int

str

...

dict

list

range

set

tuple

...



floating-point imprecision

integer overflow

1 2 3

1 2 4

1 2 5

1 2 6

1 2 7

1 2 8

1 2 9

1 2 10

1 2 9

1

1

2

0

1 3 0

9 9 9

1

9

9

0

1

9

0

0

1

0

0

0

1 0 0 0

0 0 0

1 1 1

1

1

1

0

1

1

0

0

1

0

0

0

1 0 0 0

0 0 0

integer overflow



STAR WARS

THE ORIGINAL TRILOGY





B Back
A Buy

4,000,000,000



1999

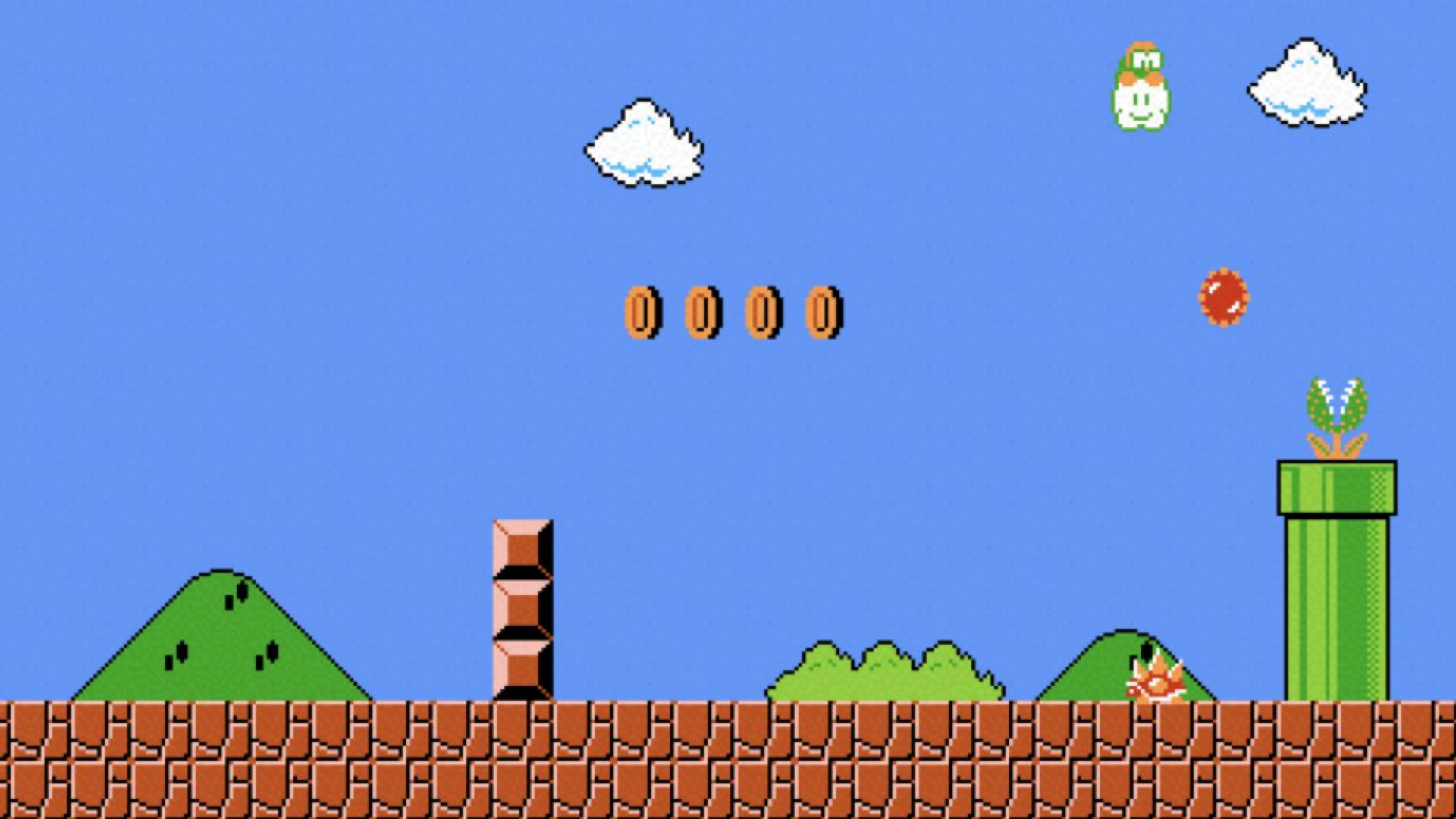
99

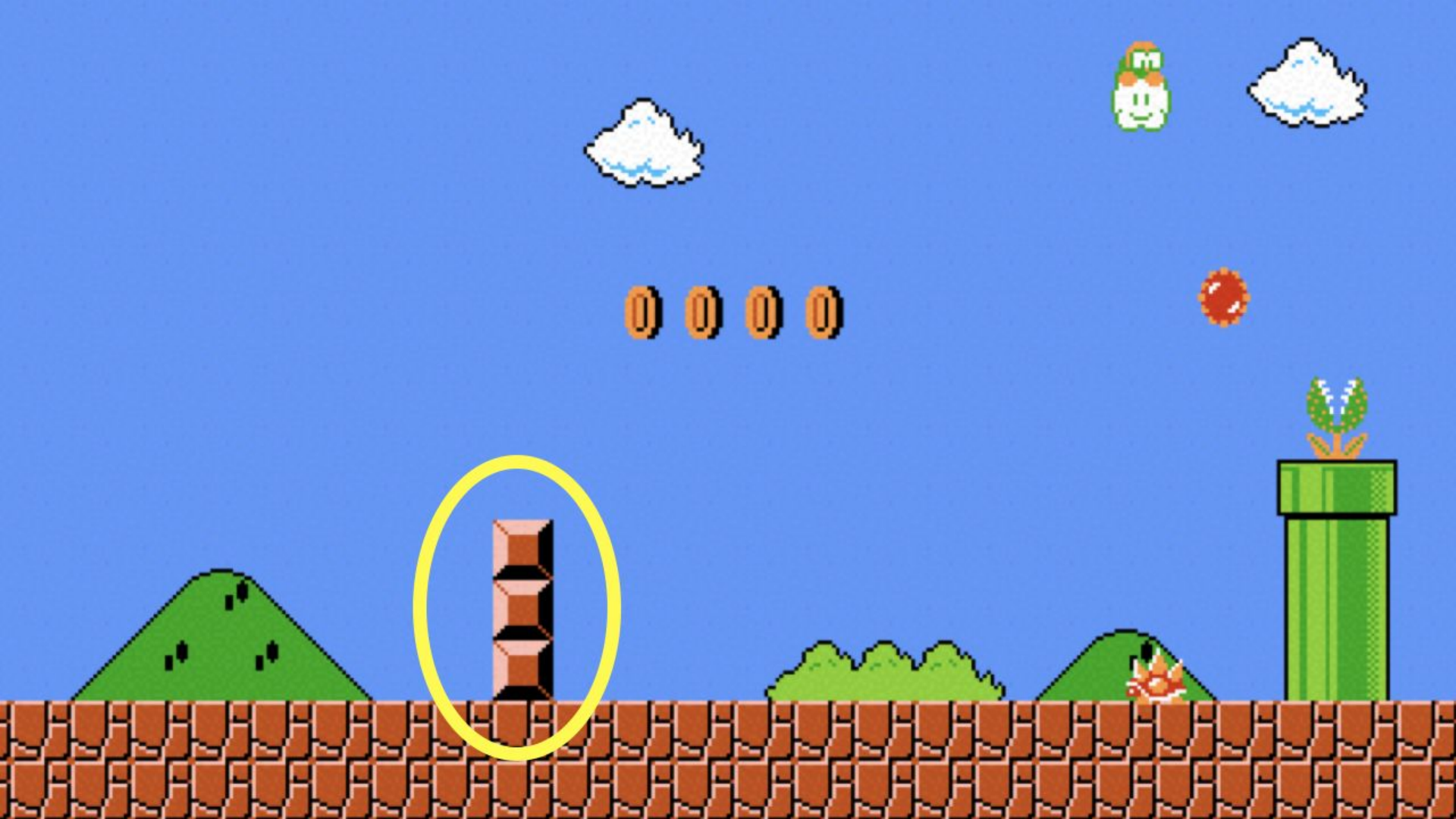
00

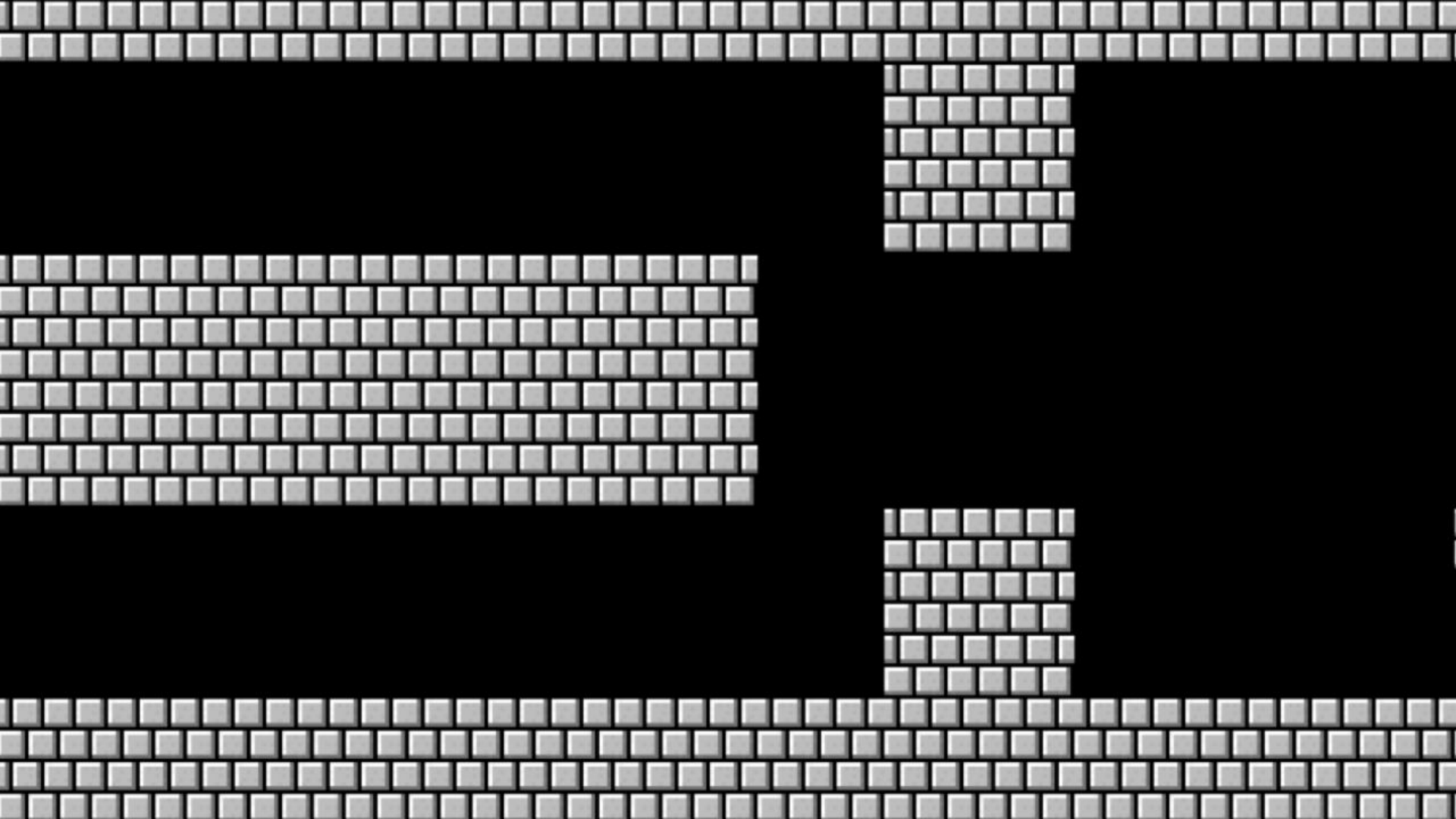


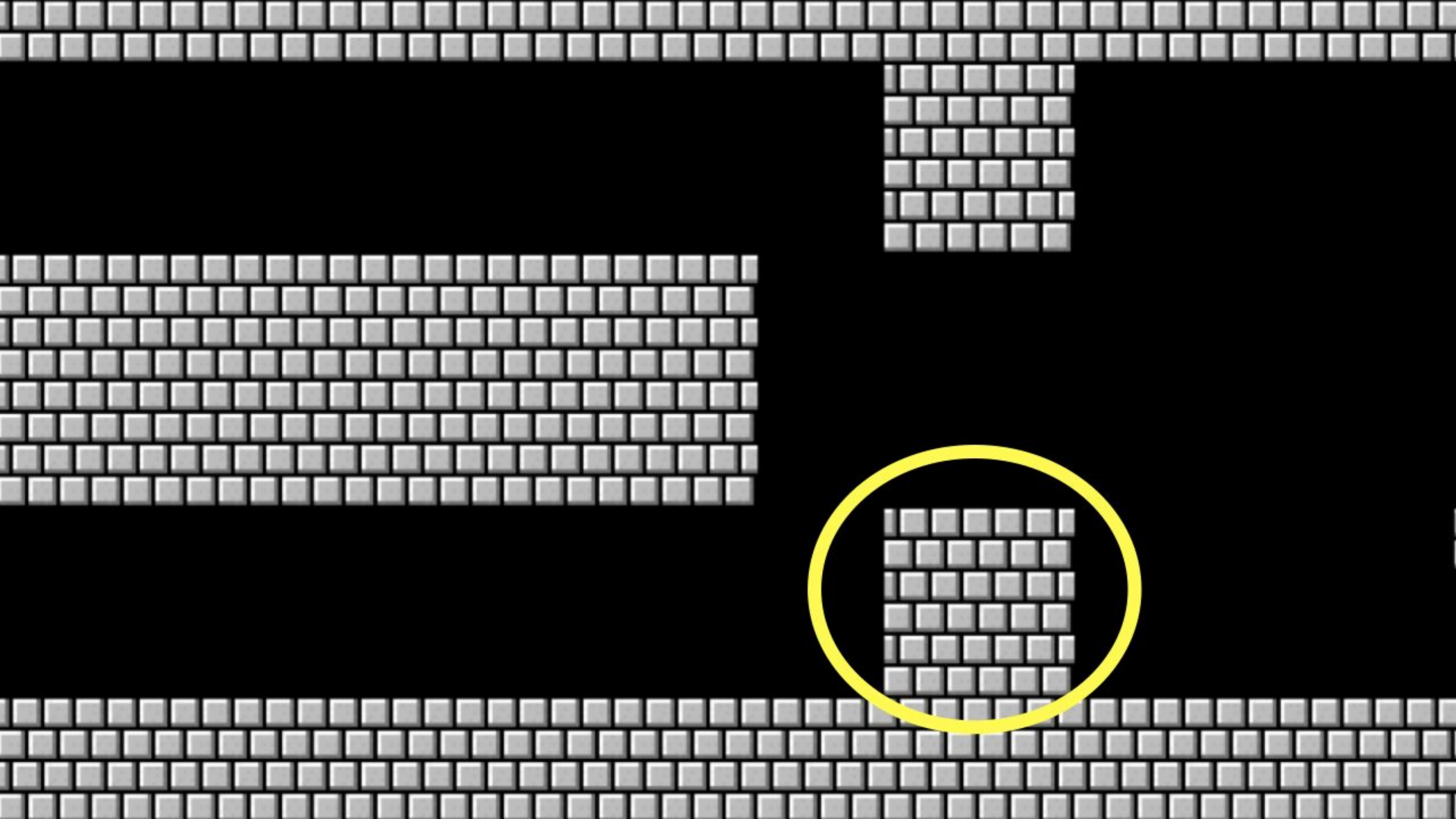












Programming Languages