
DDO loops into coding



ACADÉMIE PRODIGI

www.academieprodigi.com

Mathieu Charbonneau
Madeleine Bousquet

What is programming?

The process of creating a list of instructions that will be executed by a computer in order to accomplish a task.

Similar to when you create a recipe.



Where can we find programs?

- Robots
- Websites
- Applications
- Cars
- Hospital
- Games
- Movies (Pixar, Harry Potter)
- Televisions

How old do you need to be to
become a programmer?

I am too young, really?

Many kids already are having great career.

THOMAS SUAREZ

- Started programming at age 11
- Self-taught
- Started his own company
- Invented a 3D printer 10x faster



Want more?



A portrait of a young boy with dark hair and glasses, smiling and resting his chin on his hand. He is wearing a dark blue t-shirt with "IBM Watson" visible on it and a smartwatch on his left wrist.

TANMAY BAKSHI

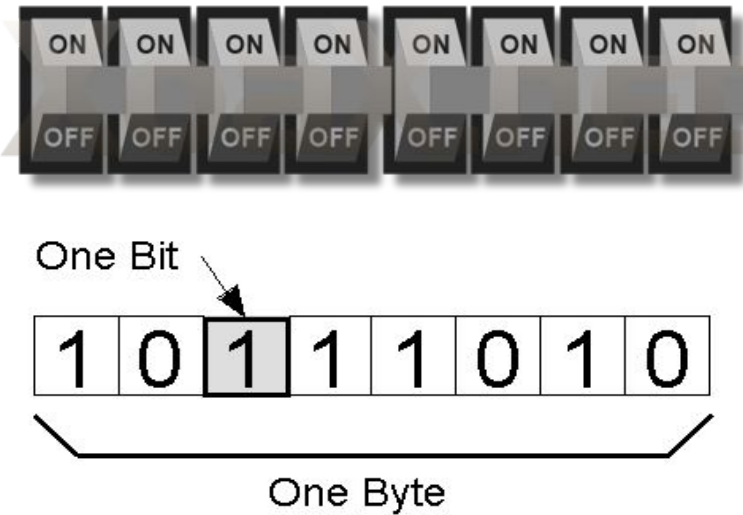
- 5yrs: Started Coding
- 9yrs: First iOS APP
- 12yrs: Youngest IBM Watson Programmer in the World.

HIRED

Google

@ 15 years old

How does a computer works?



| byte | b | 8 bits | 1 byte |
|------------|----|------------|--------------------------------------|
| kilobyte | Kb | 1024 bytes | 1 024 bytes |
| megabyte | MB | 1024 KB | 1 048 576 bytes |
| gigabyte | GB | 1024 MB | 1 073 741 824 bytes |
| terabyte | TB | 1024 GB | 1 099 511 627 776 bytes |
| Petabyte | PB | 1024 TB | 1 125 899 906 842 624 bytes |
| Exabyte | EB | 1024 PB | 1 152 921 504 606 846 976 bytes |
| Zetabyte | ZB | 1024 EB | 1 180 591 620 717 411 303 424 bytes |
| Yottabyte | YB | 1024 ZB | 1 208 925 819 614 629 174 706 176 by |
| Brontobyte | BB | 1024 YB | 1 237 940 039 285 380 274 899 124 22 |
| Geopbyte | GB | 1024 BB | 1 267 650 600 228 229 401 496 703 20 |

Binary Alphabet

ASCII Table

| Decimal | Binary | Character | Decimal | Binary | Character | Decimal | Binary | Character |
|---------|----------|-----------|---------|----------|-----------|---------|----------|-----------|
| 32 | 00100000 | space | 64 | 01000000 | @ | 96 | 01100000 | ' |
| 33 | 00100001 | ! | 65 | 01000001 | A | 97 | 01100001 | a |
| 34 | 00100010 | " | 66 | 01000010 | B | 98 | 01100010 | b |
| 35 | 00100011 | £ | 67 | 01000011 | C | 99 | 01100011 | c |
| 36 | 00100100 | \$ | 68 | 01000100 | D | 100 | 01100100 | d |
| 37 | 00100101 | % | 69 | 01000101 | E | 101 | 01100101 | e |
| 38 | 00100110 | & | 70 | 01000110 | F | 102 | 01100110 | f |
| 39 | 00100111 | ' | 71 | 01000111 | G | 103 | 01100111 | g |
| 40 | 00101000 | (| 72 | 01001000 | H | 104 | 01101000 | h |
| 41 | 00101001 |) | 73 | 01001001 | I | 105 | 01101001 | i |
| 42 | 00101010 | * | 74 | 01001010 | J | 106 | 01101010 | j |
| 43 | 00101011 | + | 75 | 01001011 | K | 107 | 01101011 | k |
| 44 | 00101100 | , | 76 | 01001100 | L | 108 | 01101100 | l |

How is a program made?

```
using System;
public class Program
{
    public static void Main()
    {
        int nombre_boucle;
        nombre_boucle = 1;
        while(nombre_boucle <= 5)
        {
            Console.WriteLine("Chiffre: " + nombre_boucle);
            nombre_boucle = nombre_boucle + 1;
        }
        Console.ReadLine();
    }
}
```

```
.class public auto ansi beforefieldinit Program
    extends [mscorlib]System.Object
{
    .method public hidebysig static void Main() cil managed
    {
        //
        .maxstack 2
        .locals init (int32 V_0,
                    bool V_1)
        IL_0000: nop
        IL_0001: ldc.i4.1
        IL_0002: stloc.0
        IL_0003: br.s      IL_0021

        IL_0005: nop
        IL_0006: ldstr     "Chiffre: "
        IL_000b: ldloc.0
        IL_000c: box      [mscorlib]System.Int32
        IL_0011: call     string [mscorlib]System.String::Concat(object,
                                                    object)
        IL_0016: call     void [mscorlib]System.Console::WriteLine(string)
        IL_001b: nop
        IL_001c: ldloc.0
        IL_001d: ldc.i4.1
        IL_001e: add
        IL_001f: stloc.0
        IL_0020: nop
        IL_0021: ldloc.0
        IL_0022: ldc.i4.5
        IL_0023: br.s      IL_0003
    }
}
```

```
01010100 01101000 01101001 01110011
00100000 01101001 01110011 00100000
01110100 01101000 01100101 00100000
01110100 01110101 01110100 01101111
01110010 01101001 01100001 01101100
00100000 01110100 01101111 00100000
01101100 01100101 01100001 01110010
01101110 00100000 01100010 01101001
01101110 01100001 01110010 01111001
00101110 00100000 01001001 00100000
01101000 01101111 01110000 01100101
00100000 01111001 01101111 01110101
00100000 01100101 01101110 01101010
01101111 01111001 00100000 01101001
01110100 00100001
```

The foundation of programming

- Events
 - When teacher says something
 - When the mouse is clicked
 - When teacher raises hand
 - When someone enters the room

The foundation of programming

- Events Say something
- Actions Scratch your head
 Touch your nose
 Stand up

The foundation of programming

- Events

Loop 3 times

- Actions

Any action

- Loops

Loop 5 times

Any action

Let's practice

When teacher says: “Are you ready?”

Loop 3 times

Say “Oh yeah!”

Let's practice

When teacher raises hand

Loop 3 times

Stand up

Sit down

The foundation of programming

- Events

- Actions

```
If it's sunny  
    Say: "Yeah"
```

- Loops

```
else  
    Say: "Oh no!"
```

- Condition

Let's practice

When teacher says: "What is the best game?"

If you think it's Minecraft

Say "Creepers!"

else if you think it's Fortnite

Touch your head

else

Cross your hands

Let's practice

When teacher clap his hands

If you wear blue clothes

Repeat 3 times

Clap your hands

else

Say "Yeah"

The foundation of programming

- Events

```
var score = 0;
```

- Actions

```
when enemy is hit
```

- Loops

```
score = score + 10
```

- Condition

- Variables

Let's practice

```
var points = 0
```

```
loop 3 times
```

```
    points = points + 2
```

```
print points
```

Let's practice

```
var points = 0
```

```
var index = 0
```

```
loop 5 times
```

```
  if index is odd
```

```
    points = points + 2
```

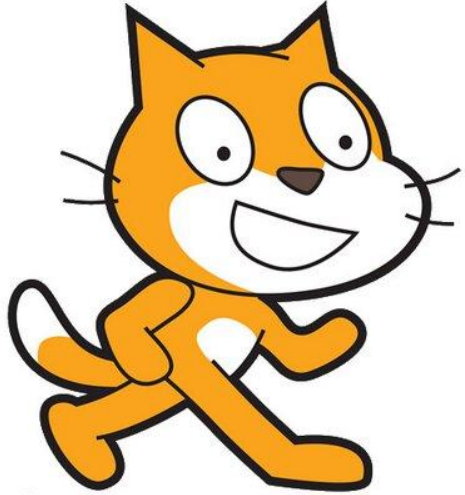
```
  else
```

```
    points = points + 1
```

```
print points
```

The foundation of programming

- Events
- Actions
- Loops
- Condition
- Variables



SCRATCH

[SCRATCH.MIT.EDU](https://scratch.mit.edu)

A solid orange horizontal bar at the bottom of the page.