

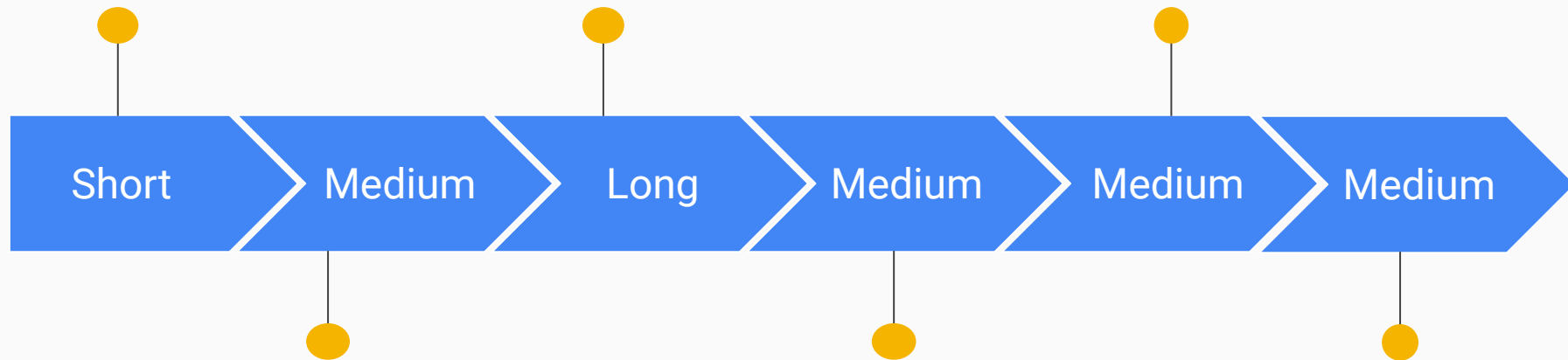
Algorithm & Javascript

& you & me & everybody

What are we
talking about ?

2 - Variables

4 - Iterations



1 - Functions

3 - Conditions

5 - Loops

programming is everywhere

- daily routines
- recipes
- scheduling
-

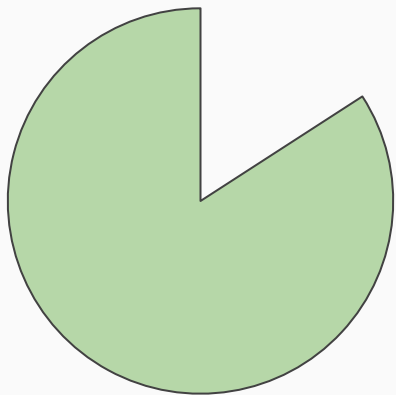
Algorithm

- Set of Instructions used to
 - Solve a problem
 - perform a computation

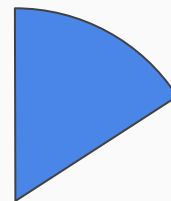
Bug

- error
 - flaw
 - failure
-
- Anything that leads to an unexpected result !

debugging code



new code



80%

coding time is spent in debugging

source: me

Global Syntax Elements For a Language

- define start and end of a program (a file, html markups)
- define start and end of a line (;)
- define comments
- operator
 - +, -, =, and, or,
- reserved word
 - if, while, function...

Before we get started...

Let's make an account on [CodePen: Online Code Editor and Front End Web Developer Community](#) to access the samples and play with them without any risk.

Let's get started

welcome to the jungle !

Syntax in Javascript

(which is common to many other languages btw)

- files or {} define **the scope** of a piece the code
- ; define the end of the instruction
 - or end of line (; is added automatically : BEWARE)
- // define a comment line
- /* define the start of a comment section
 - */ determine the end of the comment section
- operator : = == === !== && ||
- reserved word : var if

Basic program displaying something

```
<html>  
<body>  
<script type="text/javascript">  
document.write("Hi there!");  
</script>  
</body>  
</html>
```

Let's play with it

<https://codepen.io/antoine-o/pen/rNBZOje>

move code , change text, add space, carriage return, break it, use it...

- what is the use of the `<pre>` html tag ? (remove it to find out ;)
- what happen when we remove `<script>` tag ?
- what happen when we merge javascript content in a single tag ?
- what happen when we add `\n` inside the displayed text ?
- what happen when we add `
` inside the displayed text ?
- what happen when we add ' (simple quote) inside the displayed text ?
- what happen when we add " (double quotes) inside the displayed text ?
- what happen when we add \" (double quotes) inside the displayed text ?
- what happen when we add \' (double quotes) inside the displayed text ?



Functions

but no malfunctions :)

Function ?

- Has a name
- Can have input parameters
- Contains a set of instructions
- Can return an output or may modify the value in the parameters

A function is used to *factorize a piece of code* that we use several times.

This makes code smaller, *easier to maintain/develop/debug*

Simple : add 10 and return the result

```
let add10 = function(inputNumber){  
    let add10Result = inputNumber + 10;  
    return add10Result ;  
}
```


Function name

Parameter name

```
let add10 = function(inputNumber){  
    let add10Result = inputNumber + 10;  
    return add10Result;  
}
```

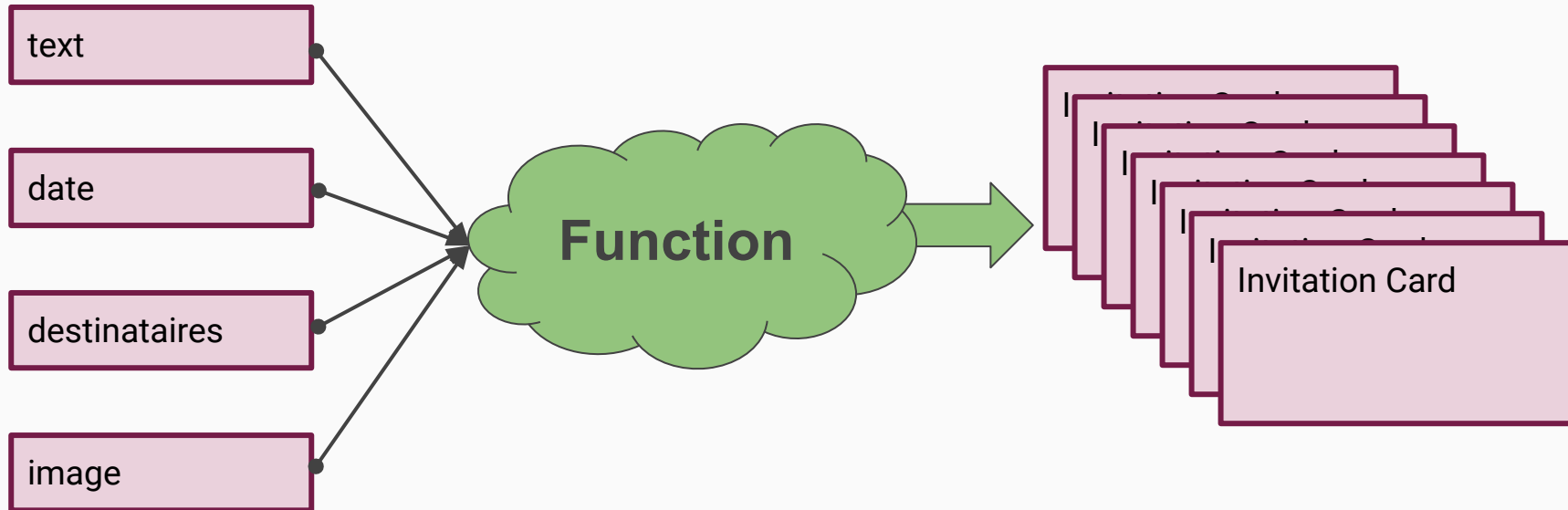
result send

Let's manipulate

<https://codepen.io/antoine-o/pen/yLBGdbq>

Uncomment each line one by one then compare the result with the previous line, try to follow the logic of the code, ask questionS :)

One step further



Variables

their use is constant

Define a variable

```
let a = 1;
```

```
var b = true;
```

```
let c = a;
```

```
let d = "let's add a number to a string into variable e : ";
```

```
let e = d + a;
```

```
let f
```

Variables are constants that can change :)

<https://codepen.io/antoine-o/pen/xxKawqq>

Manipulate / Compare a variable

operator		
=	assign a value	let a = 2
+	add a value or concatenate to a string	let b = a + 1; let c = "two equals " + a
== / !=	is equal / is not equal	1 == 1 -> true / 1 != 1 -> false 1 == "1" -> true
=== / !==	is equal strictly (same type!)	1 === 1 -> true / 1 !== 1 -> false "1" === 1 -> false
/ , * , -	divide, multiply, minus	basic math operator

Smooth Operators

<https://codepen.io/antoine-o/pen/gOYdaBE>

Complex variable - Array

index	0	1	2	3	4
value	10	"a"	0	"bcd"	"e"

```
let arrayOfRandomValues = [10,"a",0,"bcd","e"];
```

```
arrayOfRandomValues[0] // 10
```

```
arrayOfRandomValues.length // 5
```

Array (Palpatine)

<https://codepen.io/antoine-o/pen/ExYGBmJ>

Complex variable - Map

index	"a"	"b"	"c"	"d"	"e"
value	"b"	"c"	"d"	"e"	"f"

```
let arrayOfRandomValues = { "a":"b", "b":"c", "c":"d", "d":"e", "e":"f" }
```

```
arrayOfRandomValues["a"] // "b"
```

```
arrayOfRandomValues.length // 5
```

The map (is not the reality)

<https://codepen.io/antoine-o/pen/WNePoWR>

We can now structure data a bit more.

Key / Value pair makes content easier to define / read / understand

Conditions

*If your room is clean,
then you can watch TV,
(otherwise your friends won't come see you in such a dirty place)*

if and else (and cascade)

```
if (CONDITION){  
  
} else {  
  
}
```

```
if (CONDITION){  
  
} else if (CONDITION){  
  
} else {  
  
}
```

Conditions

`i > 5`

`i === 5`

`i > 5 && j < 3`

`i === j`

`i + 1 > 5`

`i + j === 5`

`i > 5 && j < 3`

`i === j`

`"" + i + j === "12"`

Conditions with $i = 1$ and $j = 4$

$i > 5$

$i + 1 > 5$

$i === 5$

$i + j === 5$

$i > 0 \ \&\& \ j < 3$

$i > 5 \ || \ j < 5$

$i === j$

$i === j$

$"" + i + j === "12"$

Imbricated Conditions with i = 5 and j = -1

`||` is OR

`&&` is AND

`!` is NOT

`()` is used for precedence

`(i + 1 > 5) && (i + j === 4)`

`!(i > 5 && j < 3)`

`i === j`

`""+i+j === "12"`

switch

```
switch (VARIABLE){  
    case value1 :  
        // do things  
        break;  
    [...]  
    default:  
        // do things  
        break;  
}
```

```
switch (VARIABLE){  
    case value1 :  
    case value2 :  
        // do things  
        break;  
    [...]  
    default:  
        // do things  
        break;  
}
```

What If ?

<https://codepen.io/antoine-o/pen/VwZgPNK>

Iterations

Please [clean your room, clean your desk, wash your teeth...] :)

for

```
for (let i = 0; i<anArrayOfThings.length ;i++){  
    // things to do  
}
```

for

```
for (let i = 0; i < ValueOfIThatExitTheIteration ;i++){  
    // things to do  
}
```

Define and set
initial value of a
counter

Stop when this
condition is false

Thing to do at the
end of a round

```
for (let i = 0; i < arrayOfThings.length; i++) {  
    // things to do in the round  
}
```

Loop

As long as your room is not cleaned, I will tell you to clean it

Based on a condition : while and do while

```
while (CONDITION){
```

```
    // things to do
```

```
}
```

```
do {
```

```
    // things to do
```

```
} while(CONDITION);
```

do once before testing the result

Infinity is near

<https://codepen.io/antoine-o/pen/bGbzgJv>

Objects

THIS is a philosophy

Complex variable - Objects

A variable with properties and functions within



Properties

- Color
- Size
- Type of screen
- Type of battery
- User ranking
- Number of sales
- Brand
- Year



Functions

- Date
- Hour
- Steps
- Heartbeat
- Set Alarm
- Unlock
- ...

Let's rock

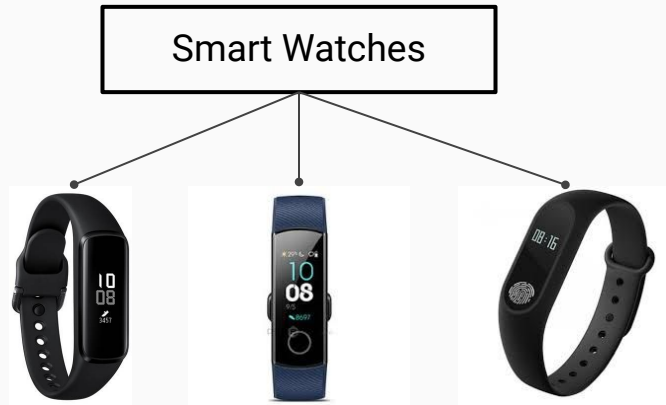
<https://codepen.io/antoine-o/pen/RwbEzgR>

Classes

Object Pattern!

Same Objects -> One Class

- Create common structure with common functions
- One piece of code for all objects of this class



Let's rock ... again and again :)

<https://codepen.io/antoine-o/pen/KKPJYdr>

Let's play !

Get out !

The idea is to display
I am out
in the document by
changing the code as
little as possible only
inside the restricted area
between

// START ALLOW EDIT
and
// END ALLOW EDIT

1pt per success
1pt extra for the best
(least changes)

1. <https://codepen.io/antoine-o/pen/JjPeeWd>
2. <https://codepen.io/antoine-o/pen/mdbQQWm>
3. <https://codepen.io/antoine-o/pen/rNBQQyg>
4. <https://codepen.io/antoine-o/pen/jONQQwv>
5. <https://codepen.io/antoine-o/pen/BaBGGwY>
6. <https://codepen.io/antoine-o/pen/LYPXXdZ>
7. <https://codepen.io/antoine-o/pen/qBWQQYN>
8. <https://codepen.io/antoine-o/pen/jONQQvd>
9. <https://codepen.io/antoine-o/pen/aboQQQd>
10. <https://codepen.io/antoine-o/pen/YzKdVWB>

Coming Next

How to interact with the document. Read value, Write Value, Change display.

All the basics that jquery rely on before using it.

[Javascript Basics - 02 - DOM Manipulation](#)

Ressources

[JavaScript | MDN](#)

[Introduction to JavaScript Course | Learn how to Build a task tracker using JavaScript](#)

[JavaScript](#)