



Cours transverse I

Houssem Ben Mahfoudh
Semestre de printemps
2020

Plan



MySQL



JavaScript

Opérateurs
Conditions
Boucles
Tableaux
Fonctions



NodeJS

MySQL



**Serveur de base de
données CUI**

10.194.69.15



PhpMyAdmin

<http://10.194.69.15/phpmyadmin>



Accès à chaque groupe:

À fournir

JavaScript

JS est un langage de programmation interprété conforme à ECMAScript.

Il est utilisé principalement dans le développement web.

C'est un langage orienté objet faiblement typé.

Introduction



Variable : Let, var



Constante : const



Type de données :

undefined

null

boolean

string

symbol

number

object

Opérateurs



Opérateur arithmétique

+(addition)
-(soustraction)
*(multiploication)
/(division)
%(modulos)
++(increment)
--(decrement)



Opérateur d'assignation

=(assignation)
+=(addition puis assignation)
-=(soustraction puis assignation)

Opérateurs



Opérateur de comparaison

==

===

!=

<

>

>=

<=



Opérateur logique

&&

||

!

Conditions

```
if(){  
    ...  
}  
else if(){  
    ...  
}  
else{  
    ...  
}
```

```
Switch(expression){  
    case x:  
        ...  
        break;  
    case y:  
        ...  
        break;  
    default:  
        ...  
}
```


Boucles

```
#while
While(condition)
{
    ...
}
```

```
# for
for( var i=0; i< 10;i++)
{
    ...
}
```

Tableaux

-
- Tableau

`let array= []`

- Ajouter un élément :

`array.push("test")`

- Supprimer le dernier élément

`array.pop()`

- Supprimer un élément

`array.splice(index,1)`

Fonctions

```
function test(a,b){  
    return a+b;  
}
```

```
let test = function (a,b){  
    Console.log(a+b);  
    return a+b;  
}
```

// ES6

```
let test = (a,b)=> a+b
```

Règles

The diagram illustrates various JavaScript formatting rules using a code snippet and red annotations with arrows. The code is as follows:

```
function pow(x, n) {  
  2 let result = 1;  
  for (let i = 0; i < n; i++) {  
    result *= x;  
  }  
  return result;  
}  
  
let x = prompt("x?", "");  
let n = prompt("n?", "");  
if (n < 0) {  
  alert(`Power ${n} is not supported,  
    please enter a non-negative integer number`);  
} else {  
  alert( pow(x, n) );  
}
```

Annotations and their corresponding rules:

- No space between the function name and parentheses**: Points to `function` and `pow`.
- A space between parameters**: Points to the space between `x` and `n` in `pow(x, n)`.
- Curly brace { on the same line, after a space**: Points to the opening brace of the function body.
- Spaces around operators**: Points to the spaces around `=`, `<`, `>`, `++`, and `*` in the code.
- A semicolon ; is mandatory**: Points to the semicolon at the end of the `return` statement.
- A space between parameters**: Points to the space between `x` and `n` in the nested call `pow(x, n)`.
- Lines are not very long**: Points to the long line of the `alert` message.
- } else { without a line break**: Points to the `} else {` block.
- Spaces around a nested call**: Points to the spaces around `pow(x, n)` in the `alert` call.
- Indentation 2 spaces**: Points to the indentation of the first line inside the function.
- A space after for/if/while...**: Points to the space after the opening brace of the `for` loop.
- An empty line between logical blocks**: Points to the empty line between the function definition and the variable declarations.

NodeJS

-
- Installer node :

<https://nodejs.org/en/download/>

- Installer SublimeText:

<https://www.sublimetext.com/>

- Installer postman:

<https://www.getpostman.com/downloads/>

- Installer Visual studio code

<https://code.visualstudio.com/>