



Structure

```
function echo(input) {
  console.log("message");
```



Types de données

```
selecting=false; e.selected=true; e.startselected=true; c.
tend(a.ui.selectable, {version: "1.8.16"})})(jQuery);
rent", axis:false, connectWith:false, containment:false,
lse,placeholder:false,revert:false,scroll:true,s
ptions; this.containerCache={}; this.element.addCl.
): this.floating=this.items.length?d.axi
is.items[0].item.css("display")):false;
bled").removeData("sortable").unbind(".
this},_setOption:func@o (d,c){if(d===
his.options[d]=c;this.widget()[c?"addCl
(atype._setOption.apply(this,arguments))
.options.disabled||this.options.typess
is, sortable-item!)==h){e=a(this);retur
(se;a(this.options.handle,e).find
rentItem=e; this._removeCurrentsFromI
 c loftl:this.helper.co
```

Typage faible

6 types différents

2 catégories

4 type de déclarations possibles

```
cting=false;e.selected=true;e.startselected=true;c.
 end(a.ui.selectable, {version: "1.8.16"})})(jQuery);
    dget("ui.sortable",a.ui.mouse,{widgetEventPrefix:
     .axis:false,connectWith:false,containment:false
            lder:false,revert:false,scroll:true
        his.containerCache={};this.elem
       floating=this.items.length?d.axi
   items[0] item.css("display")):false;
 ("sor hie").unbind("
this},_setOption:functio (d,c){if(d==
hts.options[d]=c;this.widget()[c?"addCl
otype._setOption.apply(this,arguments)
.options.disabled||this.options.typess
**sortable-item")==h){e=a(this);retu
(se;a(this.options.handle,e).
rrentItem=e; this._removeCurrentsFrom
 antentainer=this:this.refreshPos.
                    a loftl:this.helpe
```

Les types de données

Types primitifs

```
String => "message"

Number => 2 2.3

Boolean => true false

Undefined => undefined
```

Types complexes

```
Object

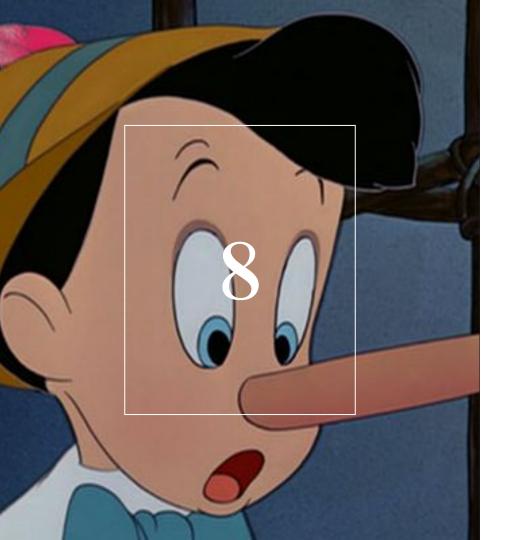
object => { prop1: true, prop2: "yes" }

null => null

array => [1, 2, 3]

Function
```

MENSONGES!!!!



JAVASCRIPT EST UN LANGAGE ORIENTE OBJET A PROTOTYPE

- Chaque type est "wrappé" par un objet
- Panel de fonctions intégrées et customisables

```
// Number
1.390.toFixed(3)
// String
"message".split()
// Boolean
true.toString()
// Array
[13, 9, 0].length
// Function
function split(arg1){}
split.arguments
```

```
selecting=false;e.selected=true;e.startselected=true;c.
tend(a.ui.selectable, {version: "1.8.16"})})(jQuery);
a.widget("ui.sortable",a.ui.mouse,{widgetEventPrefix: w
rent", axis:false, connectWith:false, containment:false,
lse,placeholder:false,revert:false,scroll:true,s
   ons; this.containerCache={}; this.element
): this.floating=this.items.length?d.axis==="x
is.items[0] item.css("display")):false;
bled").removeData("sor ab :").unbind(".
this},_setOption:func@o (d,c){if(d===
his.options[d]=c;this.widget()[c?"addCl.
(otype._setOption.apply(this,arguments))
.options.disabled||this.options.typess
is, sortable-item")==h){e=a(this);retur
(this.options.handle,e).find
rentItem=e; this._removeCurrentsFrom
 o loftl this helper.co
```

Les opérateurs

	+	-	*	1	
Number	Add	Sub	Mult	Div	
String	Concat				
Object					Access
Boolean	Add	Sub	Mult	Div	

Résultats inattendus

NaN

Infinity

```
ng=false;e.selected=true;e.startselected=true;c.
   (a.wi.selectable, {version:"1.8.16"})})(jQuery);
         ("ui.sortable",a.ui.mouse,{widgetEventPrefix:
       axis:false,connectWith:false,containment:false
            lder:false,revert:false,scroll:true
         his.containerCache={};this.elem
       floating=this.items.length?d.axi
             item.css("display")):false;
 this}, setOption: fun_ti_n(
his.options[d]=c;this.widget()[c?"add
otype._setOption.apply(this,arguments)
.options.disabled||this.options.type==
**sortable-item")==h){e=a(this);retu
(se;a(this.options.handle,e).
rentItem=e; this._removeCurrentsFro.
 repropriation this this refreshpos
                     - loftl:this.help.
```

Déclaration de variables

foo = 3

Déclaration globale

var foo = 3

Déclaration locale

let foo = 3

Déclaration au block

const foo = 3

Let + constante de référence

Attention !!! Phénomène de Hoisting

Javascript remonte automatiquement les déclarations de variables sans les initialiser en haut du scope courant pour le keyword **var**

```
selecting=false;e.selected=true;e.startselected=true;e.
tend(a.ui.selectable, {version: "1.8.16"})})(jQuery);
...idget("ui.sortable",a.ui.mouse,{widgetEventPr
rent", axis:false, connectWith:false, containment:false
lse,placeholder:false,revert:false,scroll:true,
     s; this containerCache={}; this elem
this.floating=this.items.length?d.axi
s.items[0].item.css("display")):false;
(ed").removeData("sor abi
this},_setOption:function (d,c){if(d==
his.options[d]=c;this.widget()[c?"addC
cotype._setOption.apply(this,arguments)
.options.disabled||this.options.typess
is, sortable-item")==h){e=a(this);retu
(se;a(this.options.handle,e).
rrentItem=e; this._removeCurrentsFrom.
 ***Container=this:this.refreshPosit.
                    a loftl:this.helper.
```

```
var foo = 1;

(function() {
  console.log(foo);
  var foo = 2;
  var baz = 3;
  bar = 4;
})();
Undefined
```

console.log(foo);

console.log(bar);
console.log(baz);

1 4 erreur

```
selecting=false;e.selected=true;e.startselected=true;c.
tend(a.ui.selectable, {version:"1.8.16"})})(jQuery);
3.widget("ui.sortable",a.ui.mouse,{widgetEventPrefix:
 .axis:false,connectWith:false,containment:false,
:false,helper:"original",items:">
lse,placeholder:false,revert:false,scroll:true,s
tions; this.containerCache={}; this.element.addCla
): this.floating=this.items.length?d.axis==="x"|
5. items[0].item.css("display")):false;this.offs
led").removeData("sortable").unbind(".sortable"))
this},_setOption:function(d,c){if(d===
his.options[d]=c;this.widget()[c?"addClass";
cotype._setOption.apply(this,arguments)},_mousecop
• options.disabled||this.options.type=="state.
is, "sortable-item")==h){e=a(this);return false)
(se;a(this.options.handle,e).find(",").andSelf
rrentItem=e; this._removeCurrentsFromItems
 ***Container=this; this. refreshPosition.
```

Les closures

Fonction dans une fonction

Javascript se souvient de l'environnement de chaque fonction.

Une variable qui ne devrait plus exister peut donc continuer à être appelée.

```
function creerFonction() {
  var nom = "Mozilla";
  function afficheNom() {
    console.log(nom);
  }
  return afficheNom;
}
```

var maFonction = creerFonction(); maFonction();



BLOC DE STRUCTURE

```
ecting=false;e.selected=true;e.startselected=true;c.
   d(a.ui.selectable,{version:"1.8.16"})})(jQuery);
      get("ui.sortable",a.ui.mouse,{widgetEventPrefix:
     .axis:false,connectWith:false,containment:false,
        ceholder:false,revert:false,scroll:true,
        this.containerCache={};this.elem
   his.floating=this.items.length?d.axis===
   items[0] item.css("display")):false;
  ed").removeData("so tab) ").unbind("
 this }, set Option: fun-tion +, c) {if(d==
his.options[d]=c;this.widget()[c?"addC
otype._setOption.apply(this,arguments))
.options.disabled||this.options.typess
fs, "sortable-item") == h) {e=a(this); retu
(se;a(this.options.handle,e).f
rrentItem=e; this._removeCurrentsFrom
 ***Container=this:this.refreshPosit
                     a loftl this helper
```

Les boucles

Boucle for

for(init;cond;acc) {
//Something
}

Boucle for ... in

for(x in object) {

//Something
}

Boucle do ... while

do {
//Something
} while (cond)

Boucle while

while (cond) {
//Something

Break X

Stop les itérations de X boucles

Continue

Passe à l'itération suivante

```
cting=false;e.selected=true;e.startselected=true;c.
   d(a.ui.selectable, {version: "1.8.16"})})(jQuery);
     iget("ui.sortable",a.ui.mouse,{widgetEventPrefix:
     .axis:false,connectWith:false,containment:false
             lder:false,revert:false,scroll:true,
         his.containerCache={};this.elem
       floating=this.items.length?d.axi
   items[0] item.css("display")):false;
  ed").removeData("sor :able").unbind(
 chis},_setOption:function(,c){if(d==
hts.options[d]=c;this.widget()[c?"addC
cotype._setOption.apply(this,arguments))
.options.disabled||this.options.typess
fs, "sortable-item") == h) {e=a(this); retu
(se;a(this.options.handle,e).f
rrentItem=e; this._removeCurrentsFrom
 ***Container=this:this.refreshPosi
                     c loftl: this helpe
```

Les conditions

```
if
                          Opérateurs
     If (cond) {
                                 >, >=
        //Something
                                 <, <=
     } else if (cond2) {
                                 ==
         //Something
     } else {
         //Something
switch
     switch(var) {
       case v1:
         //Something
          break:
       default:
         //Something
          break;
```



Exercice 1 Manipulation de chaînes

./exercice-1/string.js

ucfirst

1ère lettre en MAJ

hello world => Hello world

capitalize

1ère lettre de chaque mot en MAJ

hello world => Hello World

camelCase

Capitalize + coller les mots

hello world => HelloWorld

snake_case

Joindre les mots par des underscores en MIN

leet - Cryptage (uniquement les voyelles)

anaconda => 4n4cond4

A=>4, E=>3, I=>1, O=>0, $U=>(_)$, Y=>7

prop_access

"animal.type.name" => animal["type"]["name"]

prop_access(prairie, "animal.type.name")

=> praire.animal.type.name => "chien"

Si attribut non existant, afficher le chemin

jusqu'à l'attribut => "animal.gender not exist"

Si path vide ou null, renvoyer l'objet complet

verlan

Inverser chaque mot d'une phrase

Hello world => olleH dlrow

yoda

Inverser la position des mots d'une phrase

Hello world => world Hello

vig - Cryptage => Chiffre de Vigenère

wikipedia + crypto => yzixisfzy

Renvoyer une chaîne vide en cas de valeur non String



Exercice 2 Jouons avec les types

Jouons avec les types ./exercice-2/type-check.js

type_check_v1

Vérifier que le type de l'arg1 correspond à l'arg2 type_check_v1(1, "number") => true

type_check_v2

Gérer un objet conf à vérifier

type: type de l'arg1

value: valeur de l'arg1

enum: valeurs possibles de l'arg1

type_check

Gérer une conf récursive

Properties: liste des propriétés de l'objet associé à sa conf

```
{
  type: "object",
  properties: {
    propl: { type: "number" },
    prop2: { type: "string", enum: ["val1", "val2"] },
    prop3: { type: "object", properties: { prop31: "number"
  } },
    prop4: { type: "array", properties: [ "boolean"]}
}
```

Exemples

type_check_v1

```
type_check_v1(1, "number") => true
```

type_check_v2

type_check

Gérer une conf récursive

Properties: liste des propriétés de l'objet associé à sa conf

```
{
  type: "object",
  properties: {
    propl: { type: "number" },
    prop2: { type: "string", enum: ["val1", "val2"] },
    prop3: { type: "object", properties: { prop31: "number"
  } },
    prop4: { type: "array", properties: [ "boolean"]}
}
```