Réalisation de systèmes interactifs à manipulation directe en JavaScript

Ingénierie des systèmes interactifs

L2 MIASHS Parcours informatique (UE MIB0405V)

Objectifs

1. Réalisation d'un système informatique interactif
2. Réalisation d'une application web interactive
3. Une introduction au DOM et à JavaScript

Plan du cours

1. Rappel et Introduction
2. Environnement de test et débogage
3. Syntaxe et structure de Contrôle
5. Manipuler le Web avec DOM
6. Projet à faire
7. Allez Plus loin : Frameworks JS

Rappel et Intro

Impérative vs Evénementielle

Programmation **séquentielle** :

- L'exécution est bloquée tant que l'utilisateur n'a pas entré sa réponse.
- L'ordre de déroulement des instructions est toujours le même.

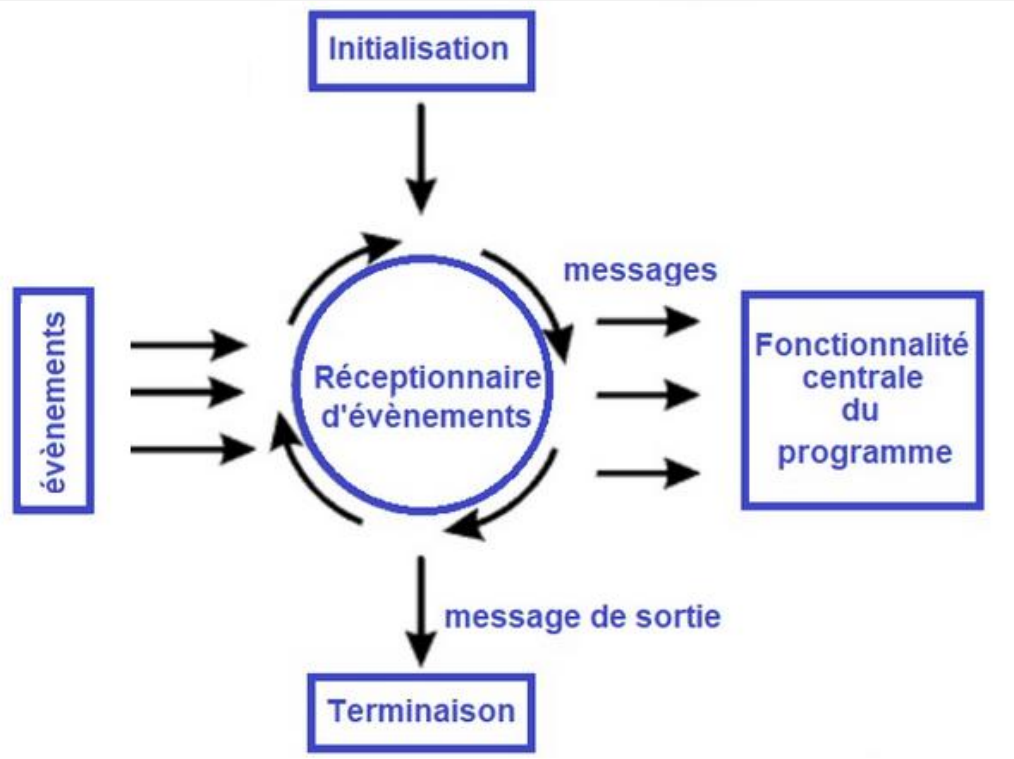
```
a=input("Enter an integer a : ")
b=input("Enter an integer b : ")
total=add(a,b)
print(total)
```

Programmation **événementielle**:

- Une boucle infinie en arrière-plan permet de détecter un événement *(getEvent())* et de le traiter *(processEvent())*.
- Le déroulement des instructions dépend de l'ordre des événements déclenchés par l'utilisateur.

```
#Boucle infinie en arriere plan
while (True):
    event = getEvent()
    processEvent(event)
```

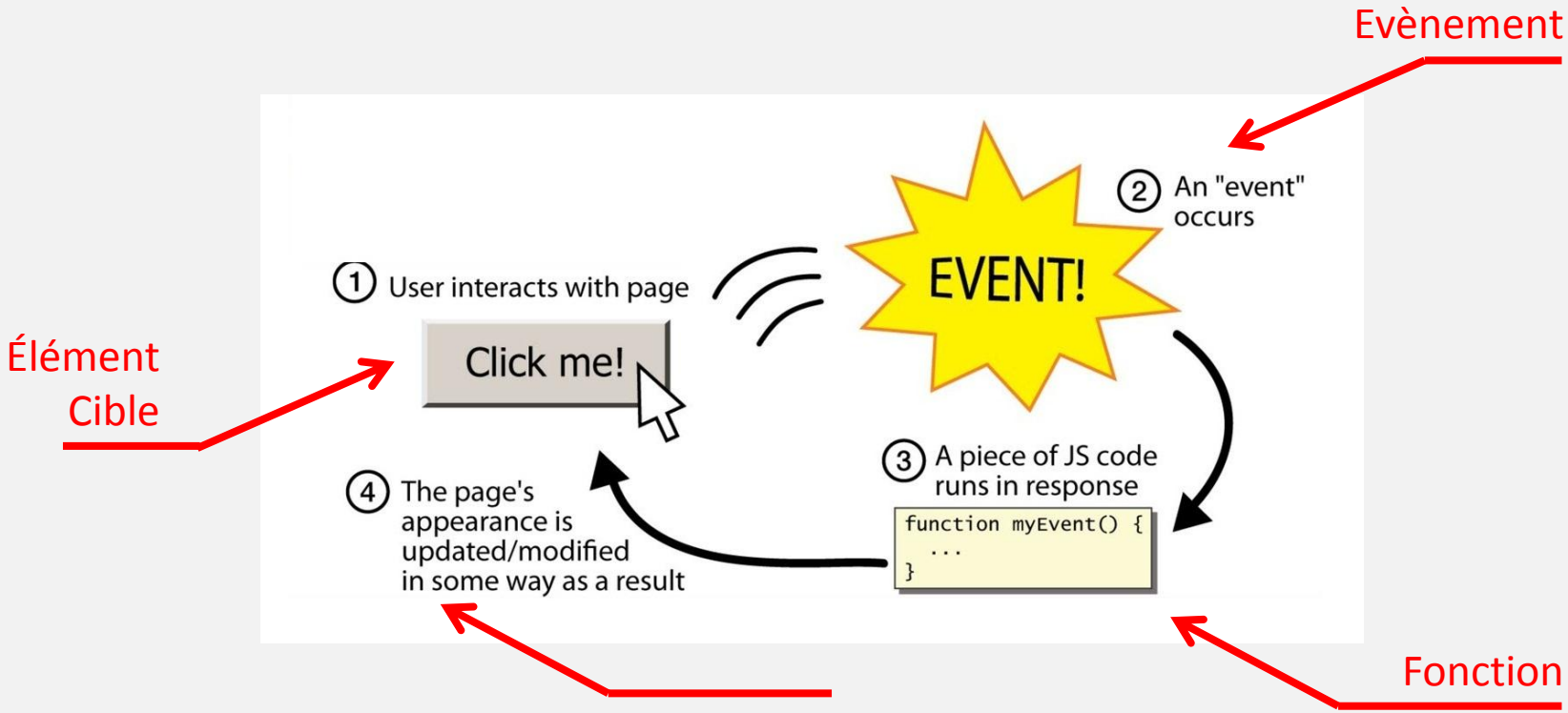

https://www.testingdocs.com/getting-user-input-in-python-3-6/
https://fgrelard.github.io/ens/even/even.pdf



- Chaque action de l'utilisateur produit des événements, exemples d'événements:
  - Souris : déplacement/glisser/cliquer, pression/relâchement du bouton
  - Clavier: pression/relâchement des touches,
  - Interruption de la minuterie (y compris les animations)
- En permanence, le programme écoute (listen) ces évènements (via des réceptionnaire, capteur) pour déclencher des fonctionnalités …

https://apcpedagogie.com/programmation-evenementielle-avec-tkinter/
https://ensweb.users.info.unicaen.fr/pres/dom-events/

Du même, en Web, la programmation événementielle repose sur l'exécution de fonctions lorsqu'un événement se produit sur élément cible



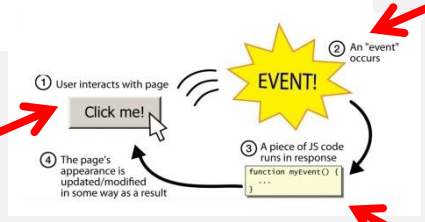
1. Ajout d'un Réceptionnaire ou capteur d'événement *(Event listener)* sur un élément
2. Le capteur surveille les événements qui se produisent sur l'élément, et déclenche une fonction quand les conditions sont remplies

https://courses.cs.washington.edu/courses/cse331/13sp/lectures/lect24-events.pdf

Exemples

Élément Cible

Evènement



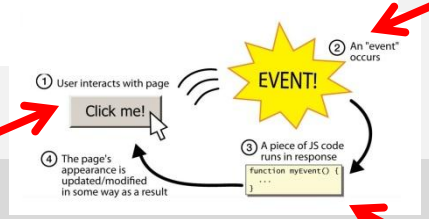
Fonction

http://www.info.univ-angers.fr/pub/stephan/LICENCE_MASS/JAVASCRIPT/COURS/interception_souris.html

Exemples

Evènement

Élément Cible



Fonction

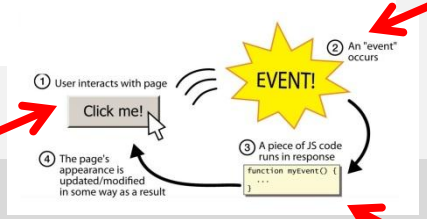
```
<HTML>
    <HEAD>
    <TITLE>Interception des événements souris</TITLE>
    </HEAD>
 <BODY>
    <H1 ALIGN=LEFT> <FONT COLOR=green> Interception des événements souris </FONT> </H1>
    <P> Exemple 01 :<BR>
    <TT>
    <A HREF= "#" ONCLICK="javascript : alert('On a cliqué !')">
    Ceci est un lien <BR>
    </A>
    </TT>
    </P>
    …
```


http://www.info.univ-angers.fr/pub/stephan/LICENCE_MASS/JAVASCRIPT/COURS/interception_souris.html

Exemples

Evènement

Élément Cible



Fonction

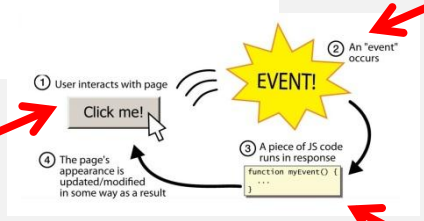
```
<HTML>
    <HEAD>
    <TITLE>Interception des événements souris</TITLE>
    </HEAD>
 <BODY>
    <H1 ALIGN=LEFT> <FONT COLOR=green> Interception des événements souris </FONT> </H1>
    <P> Exemple 01 :<BR>
    <TT>
    <A HREF= "#" ONCLICK="javascript : alert('On a cliqué !')">
    Ceci est un lien <BR>
    </A>
    </TT>
    </P>
    …
```


http://www.info.univ-angers.fr/pub/stephan/LICENCE_MASS/JAVASCRIPT/COURS/interception_souris.html

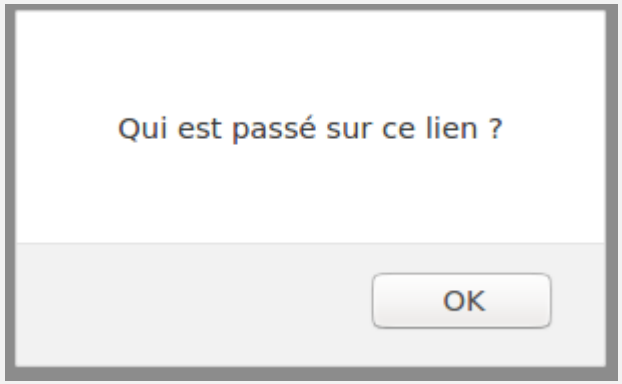
Exemples

Évènement

Élément Cible



Fonction

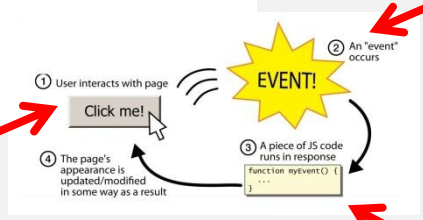


http://www.info.univ-angers.fr/pub/stephan/LICENCE_MASS/JAVASCRIPT/COURS/interception_souris.html

Exemples

Evènement

Élément Cible



Fonction

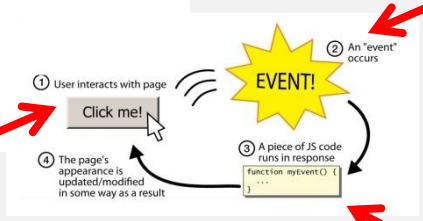
```
...
<P> Exemple 02 :<BR>
<TT>
<A HREF= "#" ONMOUSEOVER="javascript : alert('Qui est passé sur ce lien ?')">
Ceci est un autre lien <BR>
</A>
</TT>
</P>
...
```


http://www.info.univ-angers.fr/pub/stephan/LICENCE_MASS/JAVASCRIPT/COURS/interception_souris.html

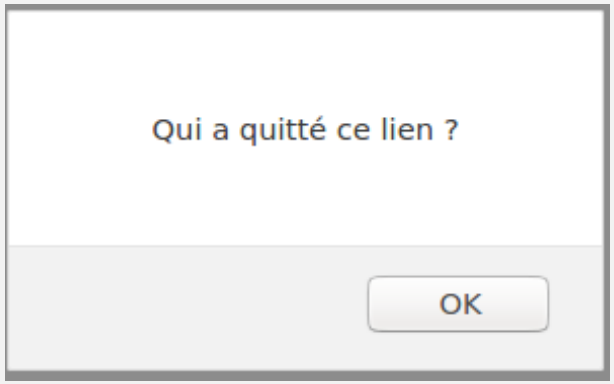
Exemples

Évènement

Élément Cible

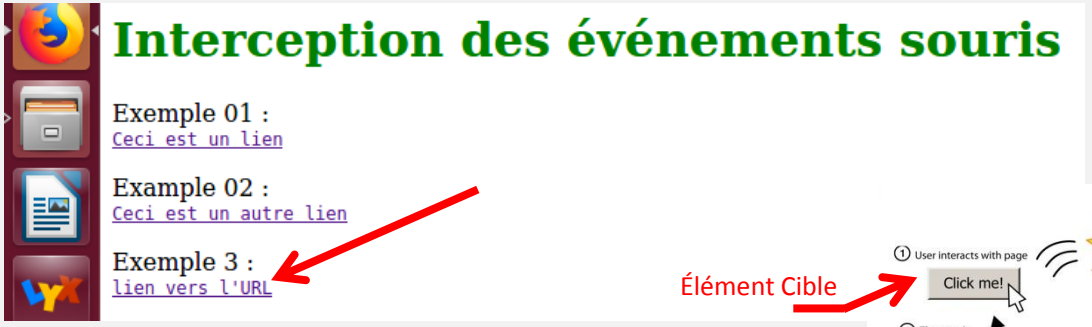


Fonction



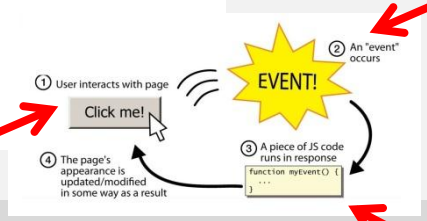
http://www.info.univ-angers.fr/pub/stephan/LICENCE_MASS/JAVASCRIPT/COURS/interception_souris.html

Exemples



Évènement

Élément Cible



Fonction

```
...
<P> Exemple 3 :<BR>
<TT>
<A HREF= "#" ONMOUSEOUT="javascript :   alert('Qui a quitté ce lien ?')">
lien vers l'URL  <BR>
</A>
</TT>
</P>...
```


http://www.info.univ-angers.fr/pub/stephan/LICENCE_MASS/JAVASCRIPT/COURS/interception_souris.html

Interception des événements souris

- En cliquant sur un lien :

```
<A HREF= "URL" ONCLICK="Code_JavaScript">
lien vers l'URL
</A>
```

- En se positionnant sur un lien :

```
<A HREF= "URL" ONMOUSEOVER="Code_JavaScript">
lien vers l'URL
</A>
```

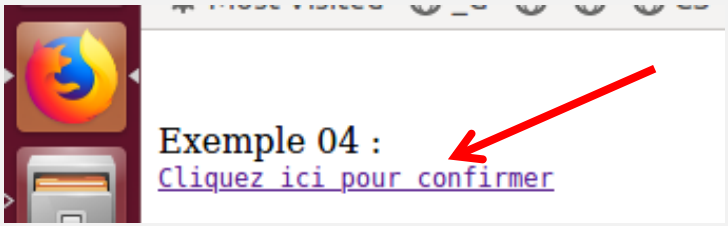
- En quittant un positionnement sur un lien

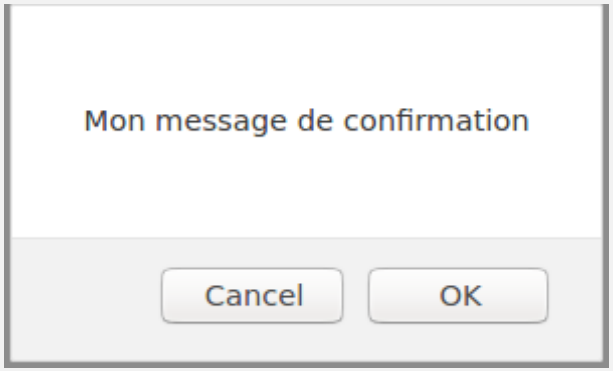
```
<A HREF= "URL" ONMOUSEOUT="Code_JavaScript">
lien vers l'URL
</A>
```

- Plus d'événements sur W3C :

www.w3schools.com/jsref/obj_mouseevent.asp

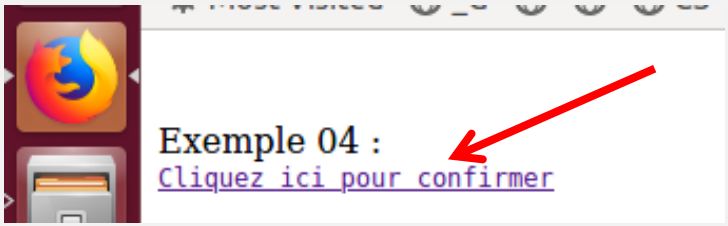
Exemples : Fenêtres préformatées





http://www.info.univ-angers.fr/pub/stephan/LICENCE_MASS/JAVASCRIPT/COURS/interception_souris.html

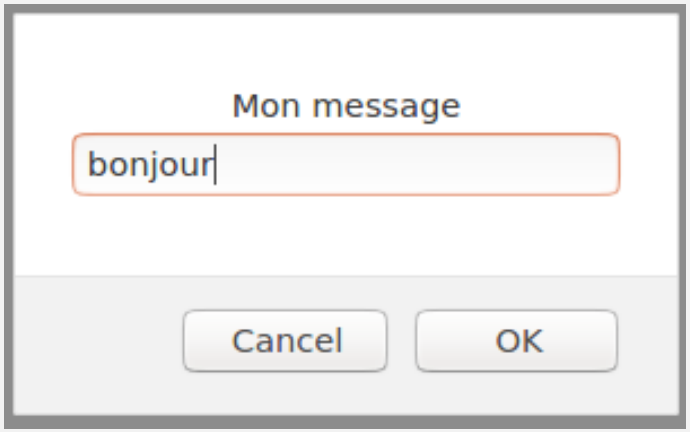
Exemples : Fenêtres préformatées




```
...
<P>Exemple 04 :<BR>
<TT>
<A HREF= "#" ONCLICK='javascript : var un_boolean = confirm("Mon message de confirmation")'>
Cliquez ici pour confirmer <BR>
</A>
</TT>
</P>
...
```


http://www.info.univ-angers.fr/pub/stephan/LICENCE_MASS/JAVASCRIPT/COURS/interception_souris.html

Exemples : Fenêtres préformatées



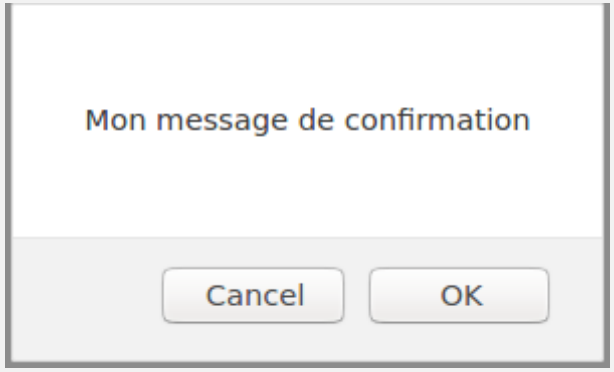
http://www.info.univ-angers.fr/pub/stephan/LICENCE_MASS/JAVASCRIPT/COURS/interception_souris.html

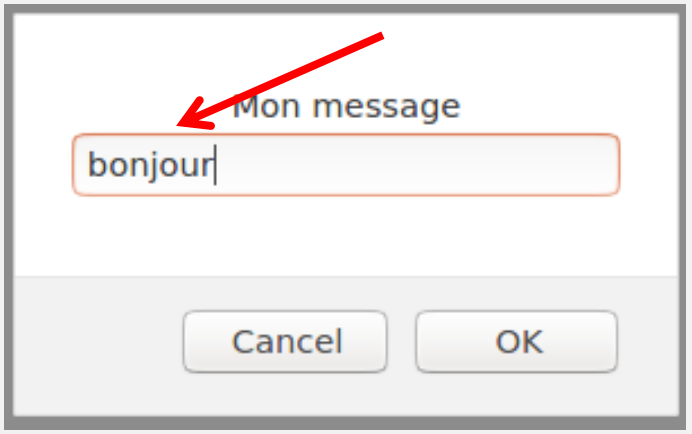
Exemples : Fenêtres préformatées

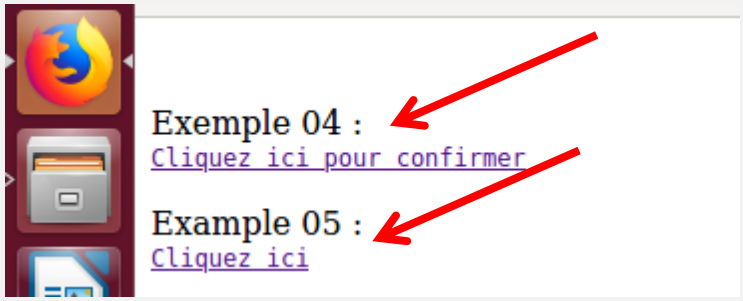

```
...
<P>Exemple 05 :<BR>
<TT>
<A HREF= "#" ONCLICK='javascript: var une_string = prompt("Mon message","Ecrivez ici"); '>
Cliquez ici <BR>
</A>
</TT>
</P>...
```


http://www.info.univ-angers.fr/pub/stephan/LICENCE_MASS/JAVASCRIPT/COURS/interception_souris.html

Manipulation …







http://www.info.univ-angers.fr/pub/stephan/LICENCE_MASS/JAVASCRIPT/COURS/interception_souris.html

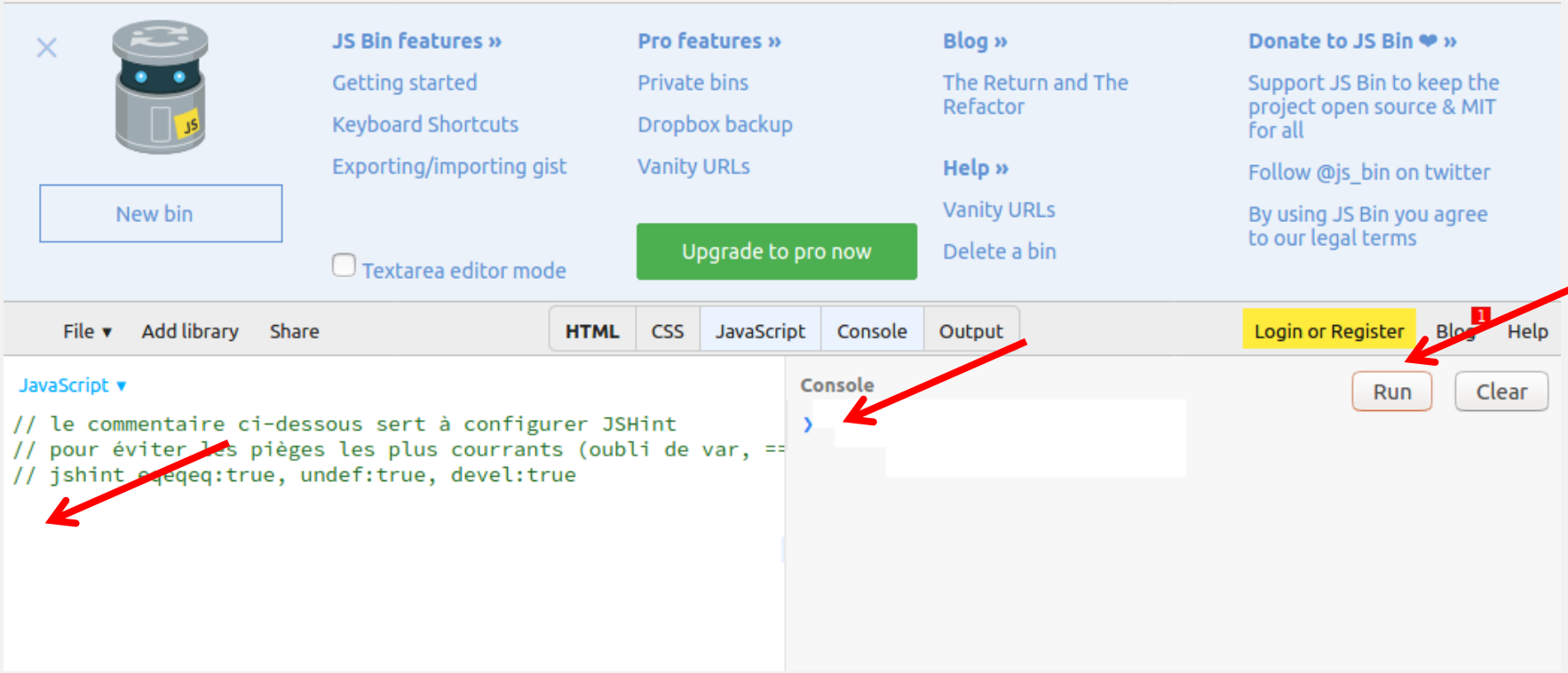
Environnement de test et débogage

https://www.lynda.com/React-js-tutorials/React-Testing-Debugging/592511-2.html

Environnement de test et débogage

- Environnement de test :

http://jsbin.com/
http://jsbin.com/fosuzu/1/edit?js,console

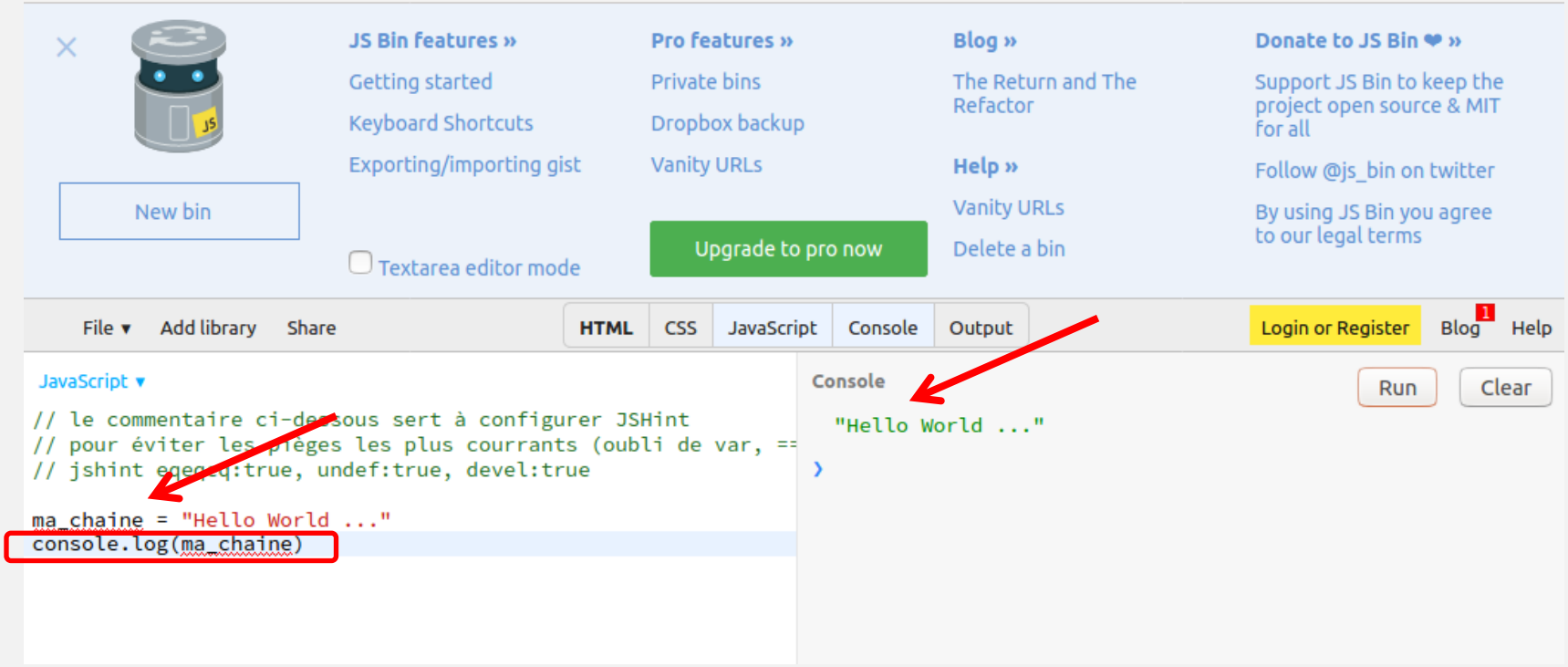


http://calcul.math.cnrs.fr/Documents/Ecoles/2010/les_bases.pdf
https://enacit1.epfl.ch/introduction-python/#dictionnaires

Environnement de test et débogage

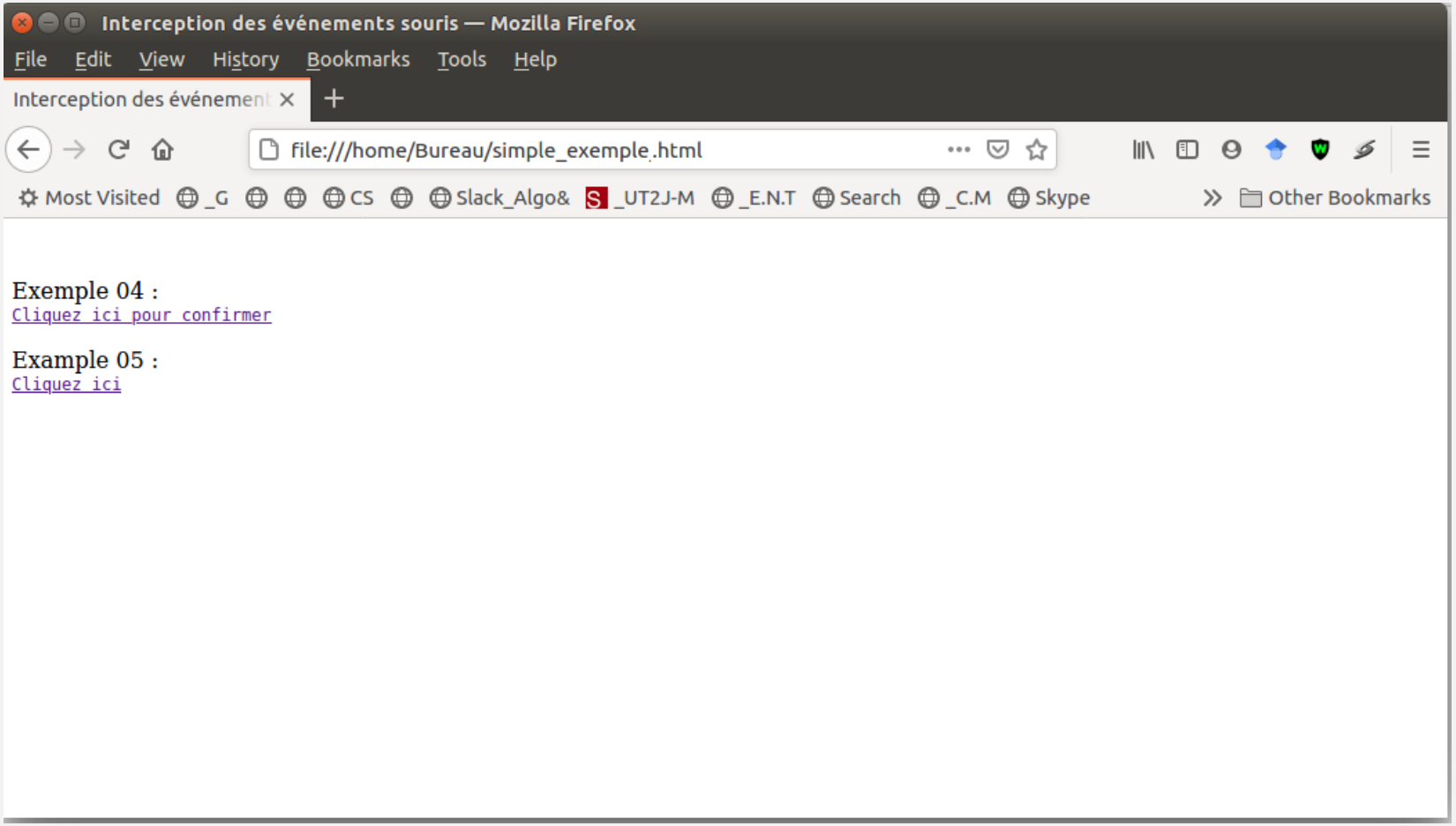
- Environnement de test :

http://jsbin.com/
http://jsbin.com/fosuzu/1/edit?js,console



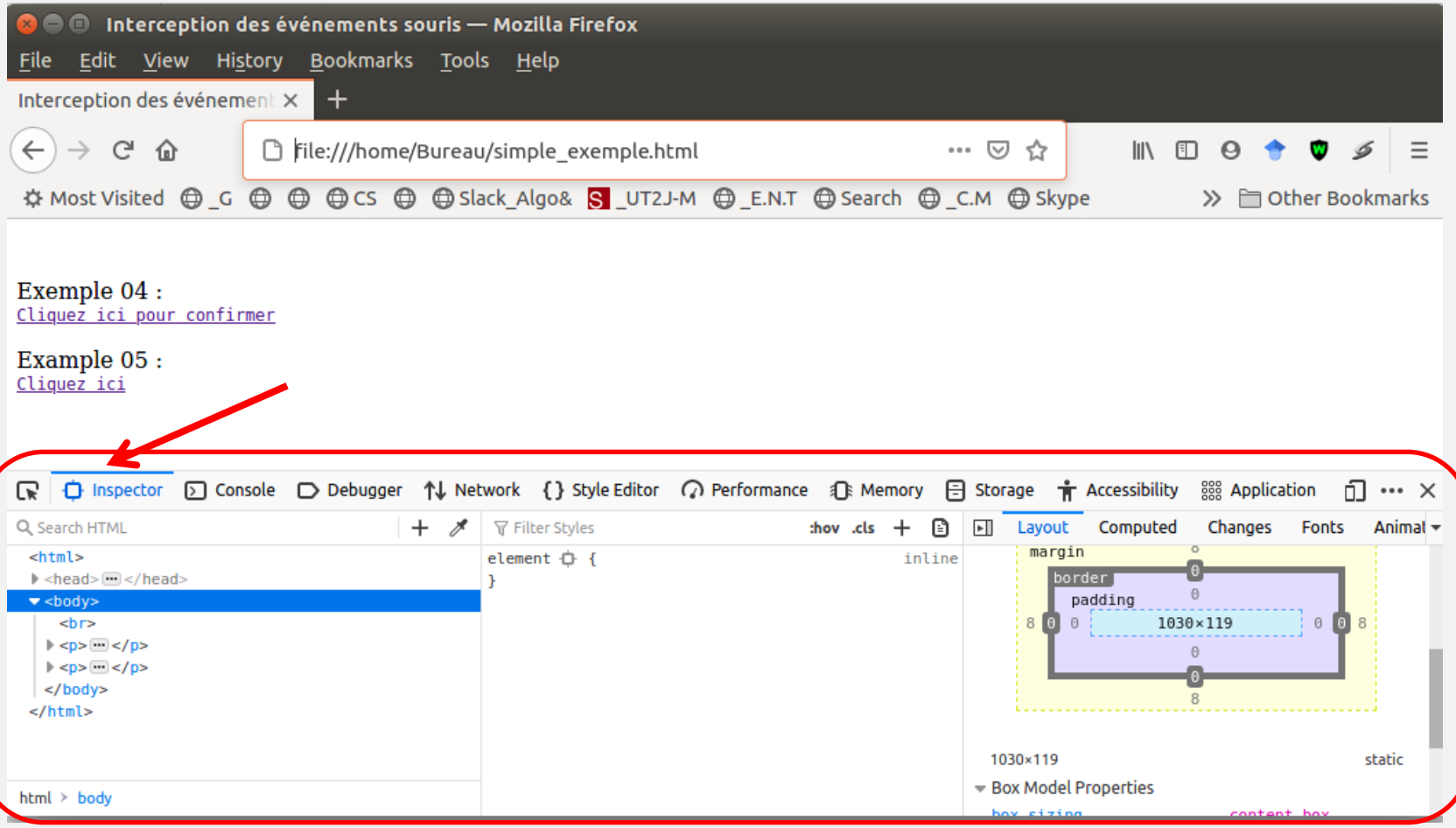
La console développeur

- Présente dans les outils de développement de Firefox ou Chrome (raccourci clavier F12* ou Cmd+Alt+J (sur Mac) ou Ctrl+Shift+J (sur PC/Window))



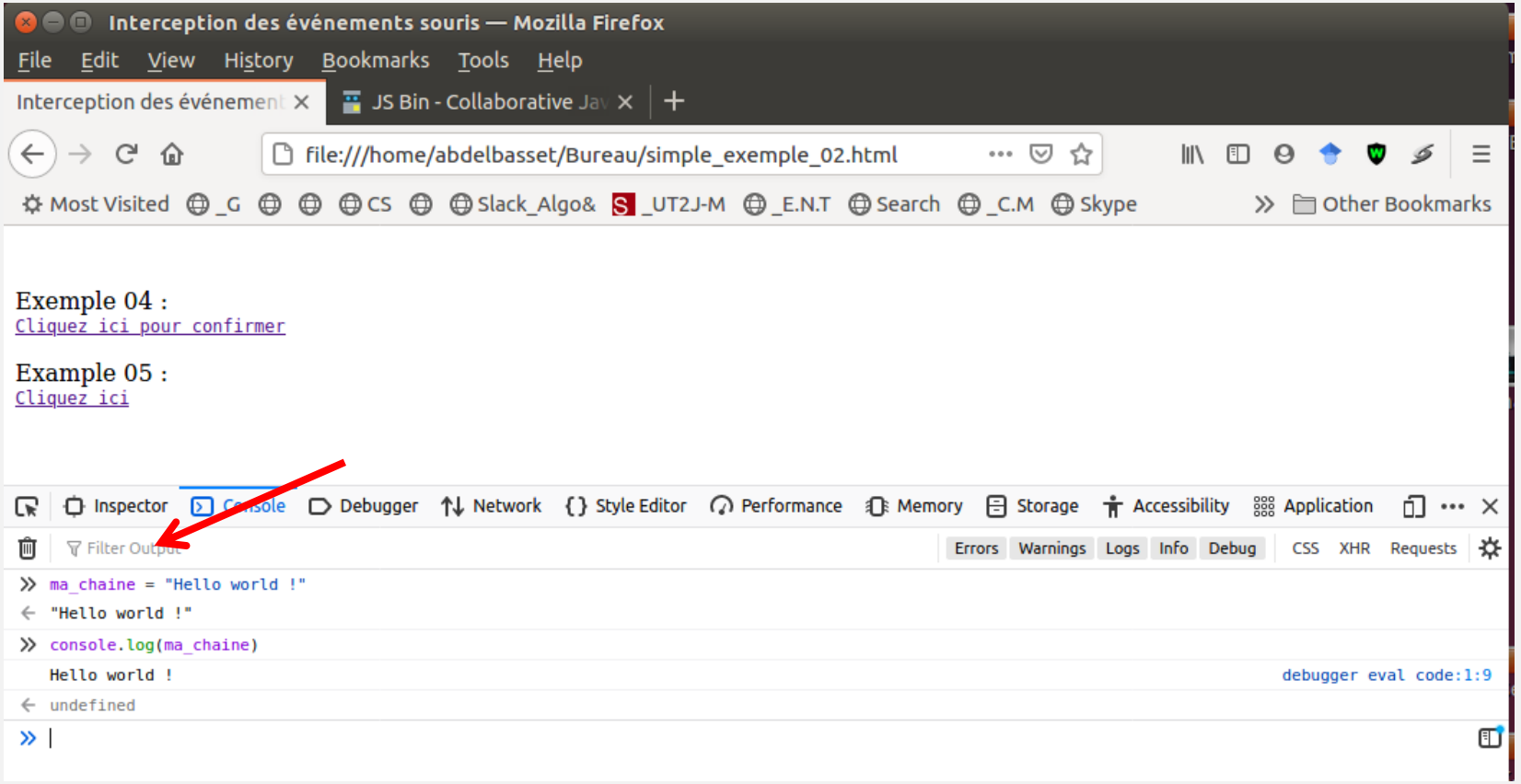
La console développeur

- Présente dans les outils de développement de Firefox ou Chrome (raccourci clavier F12)* Cmd+Alt+J (sur Mac) ou Ctrl+Shift+J (sur PC/Window)

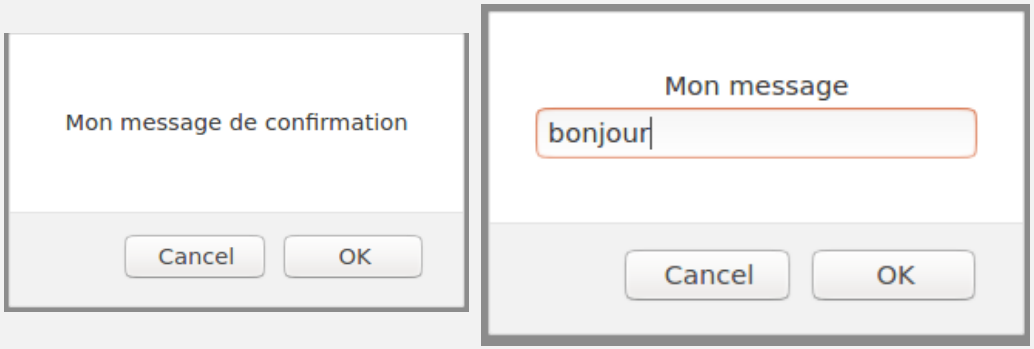


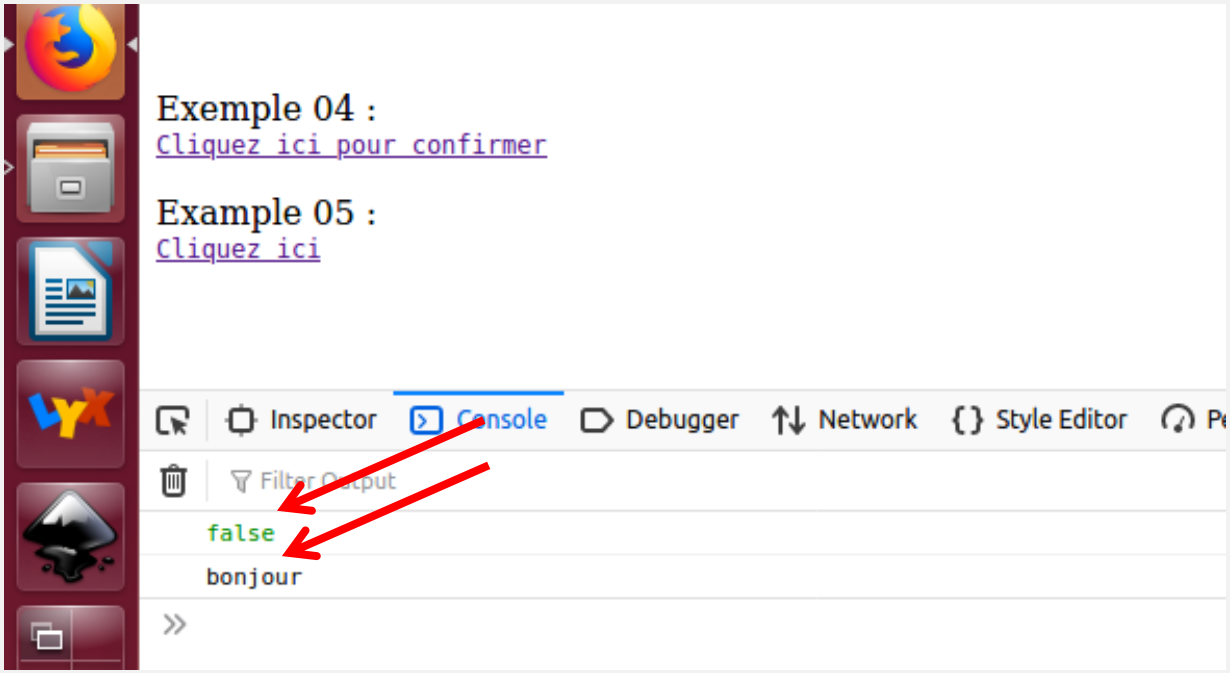
La console développeur

- Présente dans les outils de développement de Firefox ou Chrome (raccourci clavier F12)* Cmd+Alt+J (sur Mac) ou Ctrl+Shift+J (sur PC/Window)



Exemples :





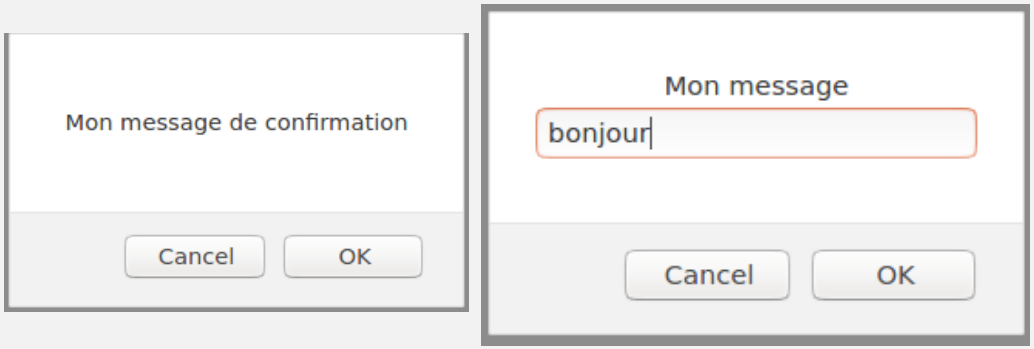
http://www.info.univ-angers.fr/pub/stephan/LICENCE_MASS/JAVASCRIPT/COURS/interception_souris.html

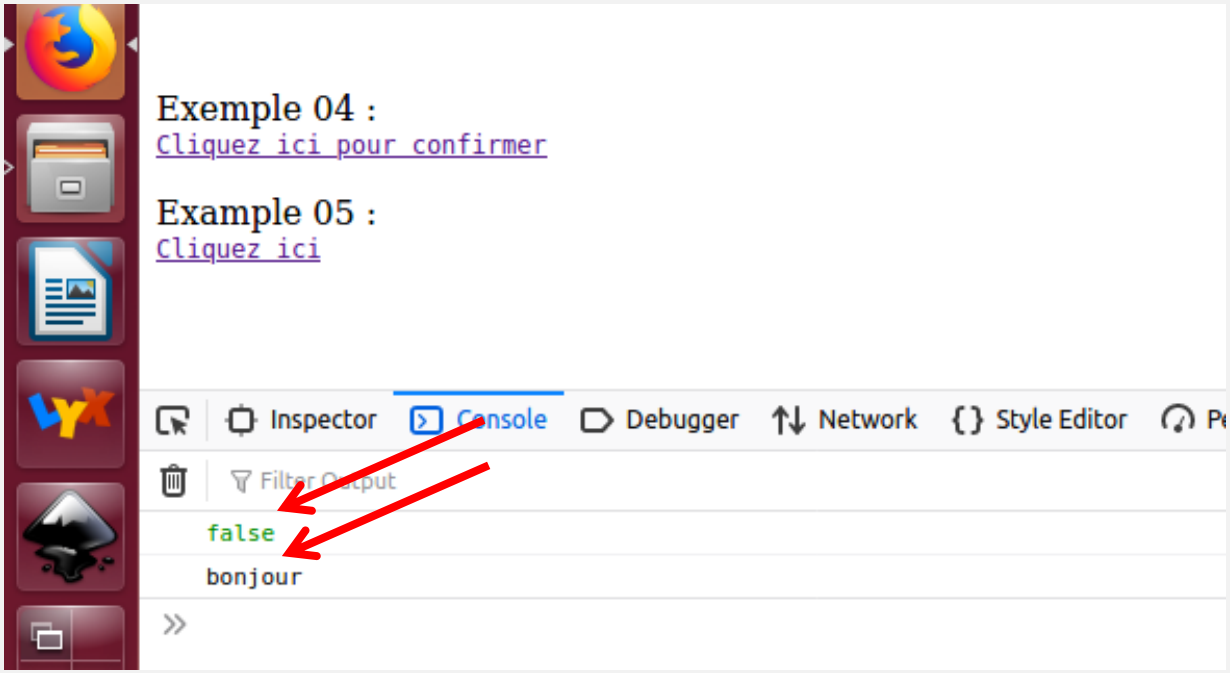
Exemples :

```
...
<P>Exemple 04 :<BR>
    <TT>
        <A HREF= "#" ONCLICK='javascript :
                              var un_boolean = confirm("Mon message de confirmation");
                              console.log(un_boolean);'>
        Cliquez ici pour confirmer <BR>
        </A>
    </TT>
</P>
<P>Exemple 05 :<BR>
<TT>
<A HREF= "#" ONCLICK='javascript :
                      var une_string = prompt("Mon message","Ecrivez ici");
                      console.log(une_string);'>
Cliquez ici <BR>
</A>
</TT>
</P>
...
```


http://www.info.univ-angers.fr/pub/stephan/LICENCE_MASS/JAVASCRIPT/COURS/interception_souris.html

Exemples :





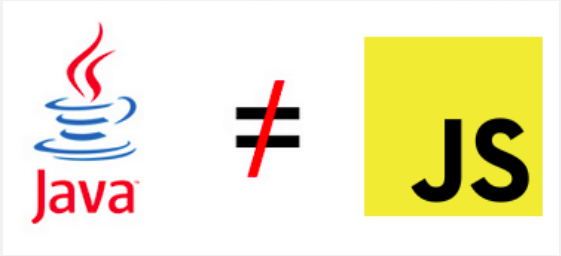
http://www.info.univ-angers.fr/pub/stephan/LICENCE_MASS/JAVASCRIPT/COURS/interception_souris.html

Syntaxe, types de base et structures de contrôle

https://www.lynda.com/Node-js-tutorials/Advanced-Node-js/661808-2.html

JavaScript (JS)

- La syntaxe de JS est inspirée de celle de Java (ou du C), la similitude s'arrête là : JS n'est pas basé sur Java.



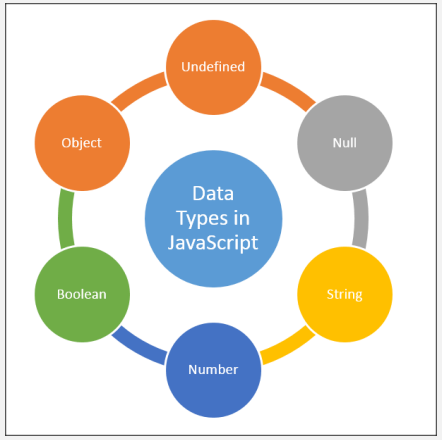
- Un langage interprété (pas de phase de compilation),
- Typage dynamique : on ne spécifie pas le type des variables ou des fonctions (comme en Python),
- Orienté Object, etc
- L'environnement d'exécution typique est le navigateur web

https://www.gchagnon.fr/cours/dhtml/evenements.html
http://calcul.math.cnrs.fr/Documents/Ecoles/2010/les_bases.pdf
https://enacit1.epfl.ch/introduction-python/#dictionnaires
http://adrienjoly.com/cours-javascript/tp01.html

Types de valeurs

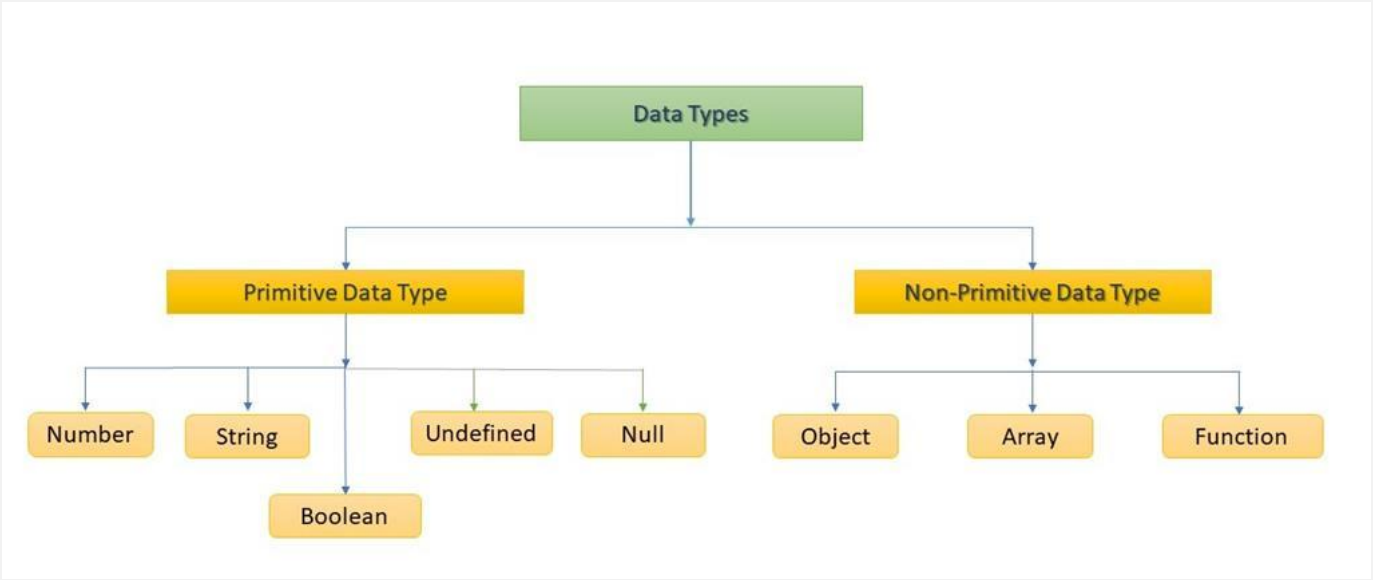
- **Types simples *(Primitifs)* :**
  - booléen *(boolean)*: true, false ;
  - nombre *(integer/float number)*: 999, 0.12, -9.99 ;
  - chaîne de caractères *(string)*: ' buzz' ;
  - types spéciaux : *null* et *undefined* qui ne peuvent recevoir aucune valeur

- **Types avancés (Objects):**
  - tableau (*array*): [1, 2, 3];
  - autres { prop: 'valeur' }, date, ;
  - etc.

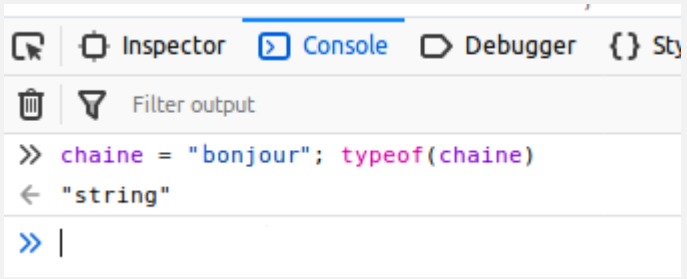


https://blog.lesieur.name/les-types-en-javascript-pour-tout-savoir/
http://adrienjoly.com/cours-javascript/tp01.html
https://www.creatingux.com/CIT230/js/datatypes.php

Opérateur Typeof()



- Le mot clé *typeof* renvoie le type de l'expression qui suit.



https://www.c-sharpcorner.com/article/datatypes-in-javascript/
http://adrienjoly.com/cours-javascript/tp01.html

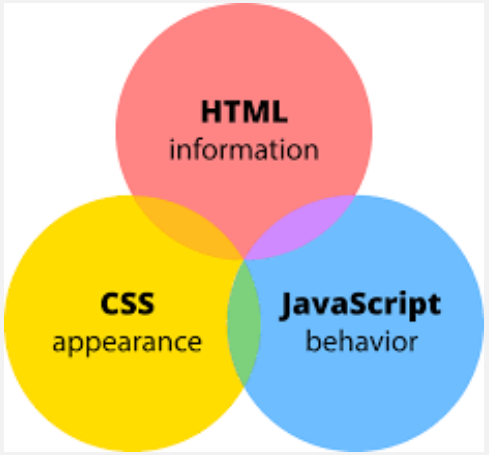
Separation of Concerns

- En informatique, Il existe un principe de conception *(design principle)* appelé separation of concerns ou *séparation des préoccupations*.

> *i.e. diviser un système en différentes parties s'occupant chacune de quelque chose de précis.*

- Dans le cadre du dev web, cela veut dire : une stricte séparation entre le fond, la forme et le comportement dynamique d'une page ===>

  - ✓ le fond est pris en charge par le code HTML,
  - ✓ la forme par la feuille de style CSS
  - ✓ et les comportements dynamiques par le JavaScript.



https://enseignement.leomartin.net/dsaa/2019-2020/HTML/1-contexte/2-separation-des-preoccupations.html
https://www.gchagnon.fr/cours/dhtml/evenements.html
https://fionacarton.com/no-css.html

Balise <SCRIPT></SCRIPT>

```
...
<P>Exemple 04 :<BR>
    <TT>
        <A HREF= "#" ONCLICK='javascript :
            var un_boolean = confirm("Mon message de confirmation");
            console.log(un_boolean);'>
        Cliquez ici pour confirmer <BR>
        </A>
    </TT>
</P>
<P>Exemple 05 :<BR>
<TT>
<A HREF= "#" ONCLICK='javascript :
            var une_string = prompt("Mon message","Ecrivez ici");
            console.log(une_string);'>
Cliquez ici <BR>
</A>
</TT>
</P>
...
```

html

Javascript

html

Javascript

html

Balise <SCRIPT></SCRIPT>

```
<!DOCTYPE html>
<html>
<head>
 <title>getElementById</title>

<script>
   function cliqueBouton (even) {
            var str = document.getElementById("myheading").innerHTML;
            alert(str);
            console.log(str);
   }
</script>

</head>
<body>
<button onclick="cliqueBouton()">Cliquez ici</button>
<h2 id="myheading">Texte quelconque</h2>
</body>
</html>
```

html

Javascript

html

Balise <SCRIPT></SCRIPT>

html

```
<!DOCTYPE html>
<html>
<head>
  <title>getElementById</title>
<script> src="js/monscript.js"</script>
</head>
<body>
<button onclick="cliqueBouton()">
Cliquez ici
</button>
<h2 id="myheading">Texte
    quelconque</h2>
</body>
</html>
```

Javascript

```
function cliqueBouton (even) {
    var str = document.getElementById
              ("myheading").innerHTML;
    alert(str);
    console.log(str);
}
```

Structures de contrôle : Conditions

```
if (expression) {
      // si expression == true,
} else {
      // sinon,
}
```

```
var  age  =  prompt ('Entrez votre Age svp !')
if (age > 18) {
            alert('Bonjour, vous êtes majeur !')
} else {
            alert('Bonjour, vous êtes mineur')
}
```

Test d'égalité:

- En JS, on teste l'égalité avec l'opérateur ===, et l'inégalité avec l'opérateur !== :

```
"42" === 42 // est faux
```

- == (et !=) ont une sémantique différente, ils considèrent que deux valeurs de types différents sont égales si on peut passer de l'une à l'autre par conversion de type. Par exemple :

```
"42" == 42 // est vrai
```

https://perso.liris.cnrs.fr/pierre-antoine.champin/enseignement/intro-js/s1.html
https://adrienjoly.com/cours-javascript/tp03.html

Boucles

```
For      ( /* initialisation */ ; /* condition */ ; /* incrémentation */ ) {
        // instructions à répeter */
}
```

```
For ( i=0 ; i<10 ; i+=1) {
   j = j*i;
}
```

```
while  (i < 10)   {
        j = j * i ;
        i  += 1;
}
```

```
console.log(1);
console.log(2);
console.log(3);
```

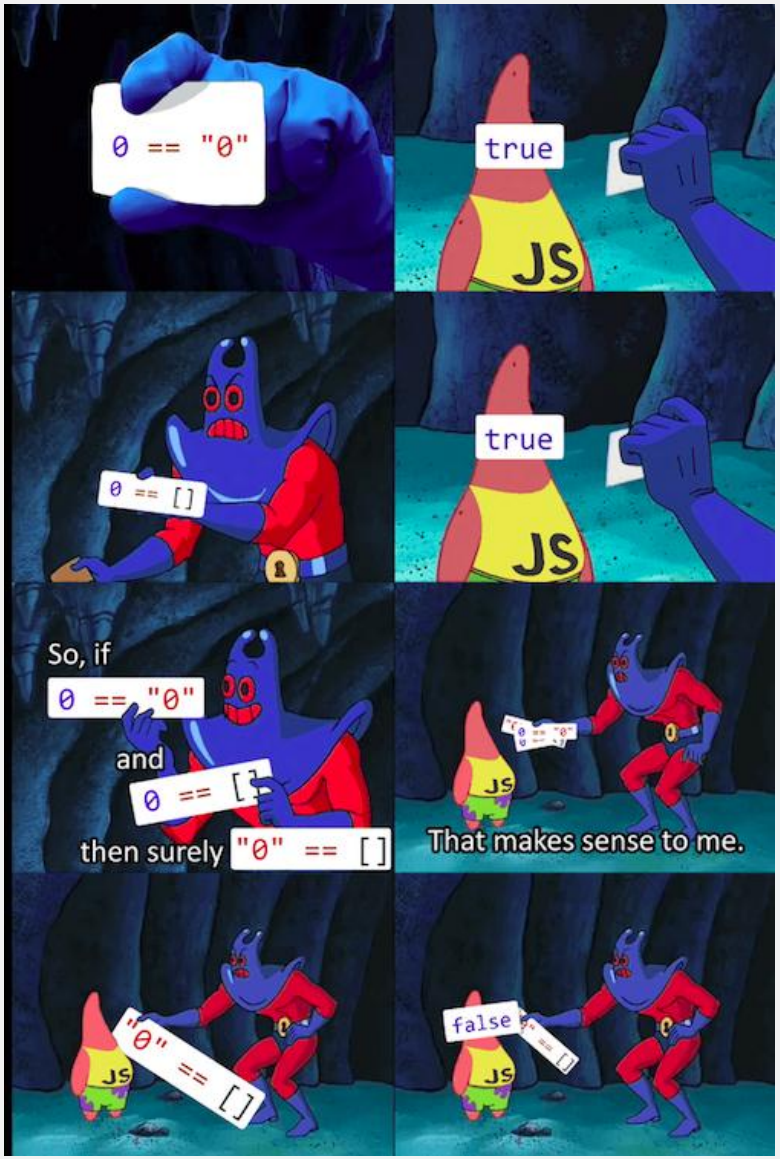
```
for ( var monNombre = 1; monNombre <= 3; monNombre++ ) {
        console.log( monNombre );
}
```

https://perso.liris.cnrs.fr/pierre-antoine.champin/enseignement/intro-js/s1.html
https://adrienjoly.com/cours-javascript/tp03.html

Manipulation … Fizz Buzz

- Écrire un programme qui affiche les nombres de 1 à 199 (compris) dans la console.
  - pour les multiples de 3, afficher ***Fizz*** au lieu du nombre.
  - pour les multiples de 5, afficher ***Buzz*** au lieu du nombre.
  - pour les nombres multiples de 3 et 5, afficher uniquement ***FizzBuzz***.

https://adrienjoly.com/cours-javascript/tp03.html
https://morioh.com/p/97ff688ca699



https://www.freecodecamp.org/news/explaining-the-best-javascript-meme-i-have-ever-seen/