

PARTIE 4 :

JQUERY

CHAPITRE 1 : FONDATIONS



WRITE LESS, DO MORE.



- Des outils tout fait
- Une compatibilité avec différents navigateurs
- JQuery, du JavaScript qui produit du JavaScript ?

JQUERY

- C'est un API JavaScript
- Un fichier de 10 000 lignes
- Un fichier compressé d'une ligne

 jquery.js	Finished	script	index.html	299 kB	7 ms
 jqueryMin.js	Finished	script	index.html	89.5 kB	7 ms

- On privilégiera donc la version compressé pour la production
- Une version slim et slim compressé (moins de fonctionnalités)
- <https://jquery.com/download/>
- C'est OpenSource
 - La version non compressée peut présenter des avantages pour le développeur

JQUERY

- Le fichier HTML est la fondation du site
 - Il engendre le chargement de ressources
 - Un ou des fichiers CSS qui appliquent des changements à la page
 - Un ou des fichiers JavaScript
 - permettant à l'utilisateur d'activer des changements sur la page de manière dynamique
 - permettant au développeur de programmer des changements en fonction de certains paramètres
 - Dans cet écosystème, le JQuery n'agit pas directement sur le fichier HTML, mais plutôt via votre fichier JavaScript
 - Par conséquent il faut charger le fichier **JQuery** avant le fichier **JavaScript**

JQUERY

- En 2012 plus de la moitié des sites utilisent JQuery d'après une étude de W3techs
- Les nouveaux frameworks tels que Node.js ou React peuvent changer la donne
- Le JQuery reste une technologie à connaître pour la réalisation de site web de petite échelle

APPORTS DU JQUERY

- Simplifier la syntaxe d'accès au DOM

```
2 //JavaScript classique
3 var madiv=document.getElementById("madiv");
4
5 //JQuery
6 var madiv=$("#madiv");
```

- Fournir une panoplie d'outils pour faciliter le dynamisme d'une page HTML

```
5 //JQuery
6 = $(document).ready(function(){
7   = $("#madiv").click(function(){
8     $(this).fadeIn();
9   });
10 });
```

VS

```
14 = function fadein(element) {
15   var op = 0.1; // initial opacity
16   element.style.display = 'block';
17   = var timer = setInterval(function () {
18     = if (op >= 1){
19       clearInterval(timer);
20     }
21     element.style.opacity = op;
22     element.style.filter = 'alpha(opacity=' + op * 100 + ")";
23     op += op * 0.1;
24     }, 10);
25 }
```

APPORTS DU JQUERY






- Une librairie que tout le monde utilise, c'est une librairie fiable, qui bénéficie de la participation d'une vaste communauté.
- Une myriade de plug-in
- Une librairie extensible
 - Créer ses sélecteurs
 - Etendre les fonctionnalités existantes
 - Créer des plugins

SYNTAXE

- Le dollar → `$()`
- Le jQuery → `jQuery()`
- Ce sont les mêmes, des alias
 - Ils prennent en paramètre un sélecteur css
- Pour faciliter les exemples dans le dom je code dans
 - La fonction anonyme
 - Passé en paramètre de l'événement ready
 - De l'objet document

```
1
2 $(document).ready(function() {
3
4   console.log(jQuery("#madiv").attr("id"));
5   console.log($("#madiv").attr("id"));
6
7 });
```



		top			Filter	Default levels	
		madiv				javascript.js:3	
		madiv				javascript.js:4	
		>					

OBJET JQUERY

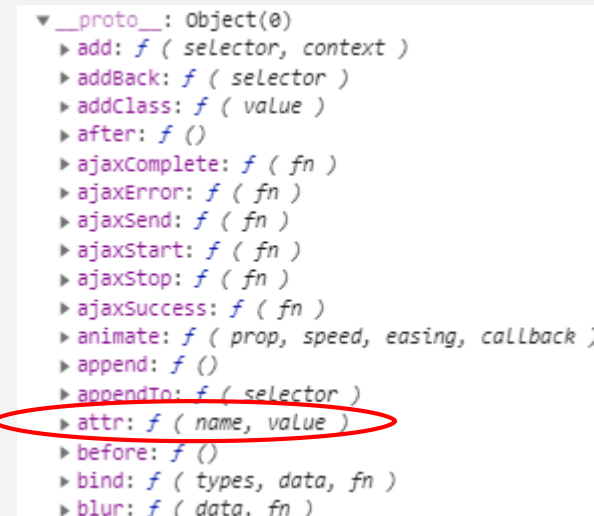
- /\ l'objet retourné par l'accès en JQuery n'est pas le même qu'en JavaScript classique

```
2 //JavaScript classique
3 var madiv=document.getElementById("madiv");
4
5 //JQuery
6 var madiv=$("#madiv");
```



- C'est un objet JQuery, composé de toutes les propriétés et les méthodes permettant leur manipulation, on accède à celles-ci avec un **point**

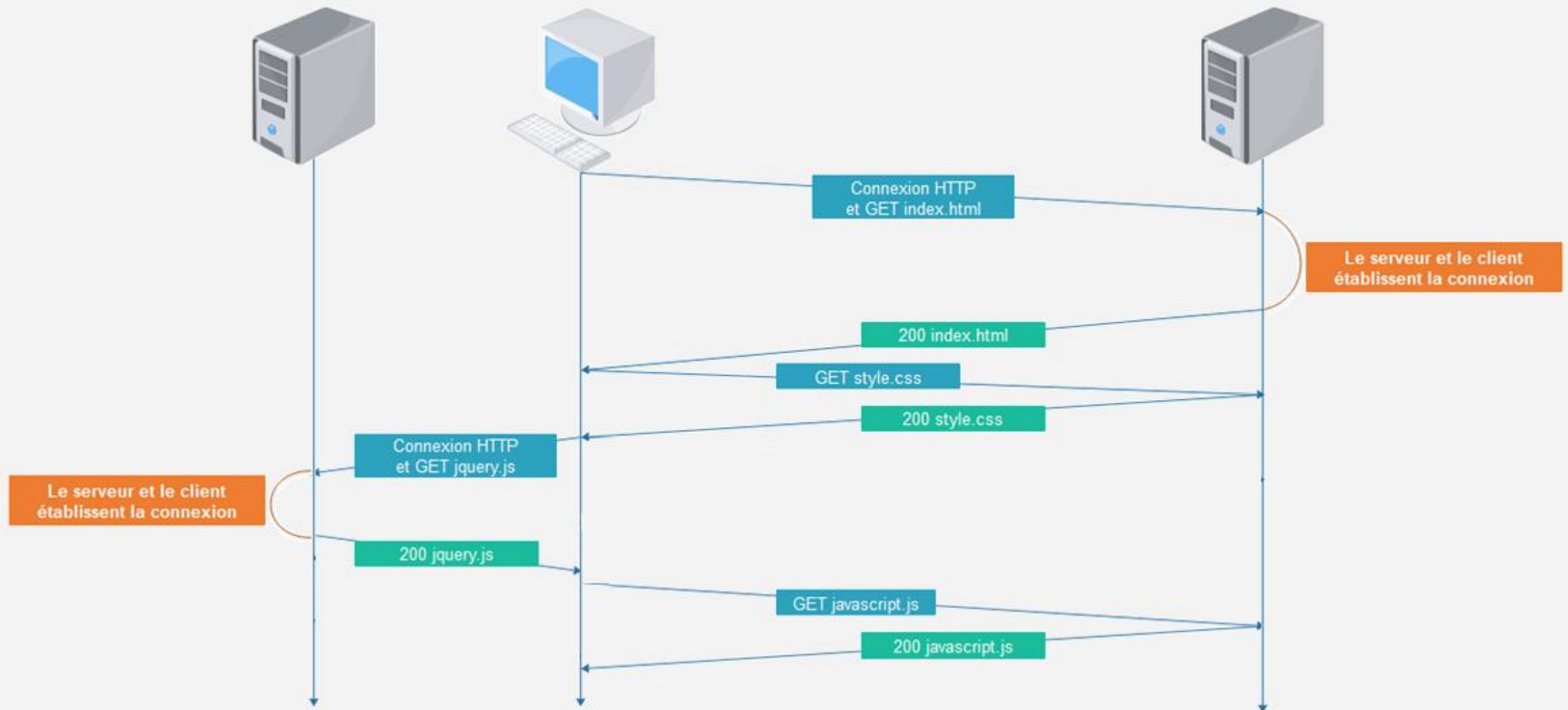
```
3
4 console.log($("#madiv").attr("id"));
```



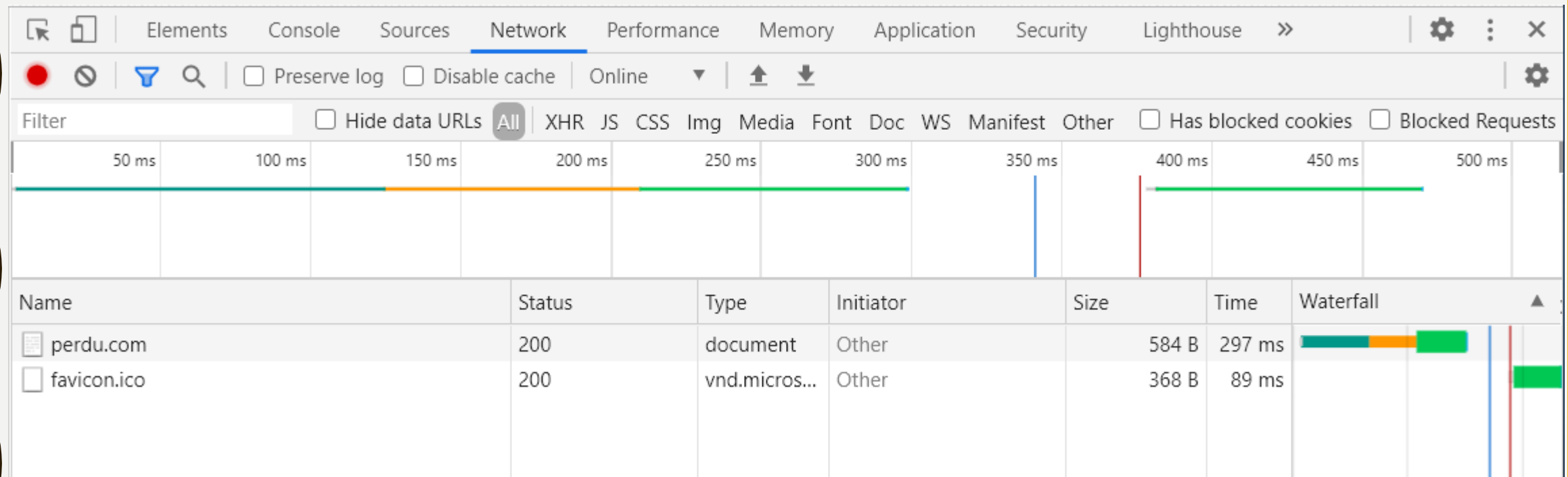
INCLURE JQUERY

- Soit on télécharge le fichier jquery pour l'intégrer à notre dossier
- Soit on fait appel à un CDN
 - <https://code.jquery.com/>
- CDN pour Content Delivery Network (réseau de distribution de contenu)

JQUERY

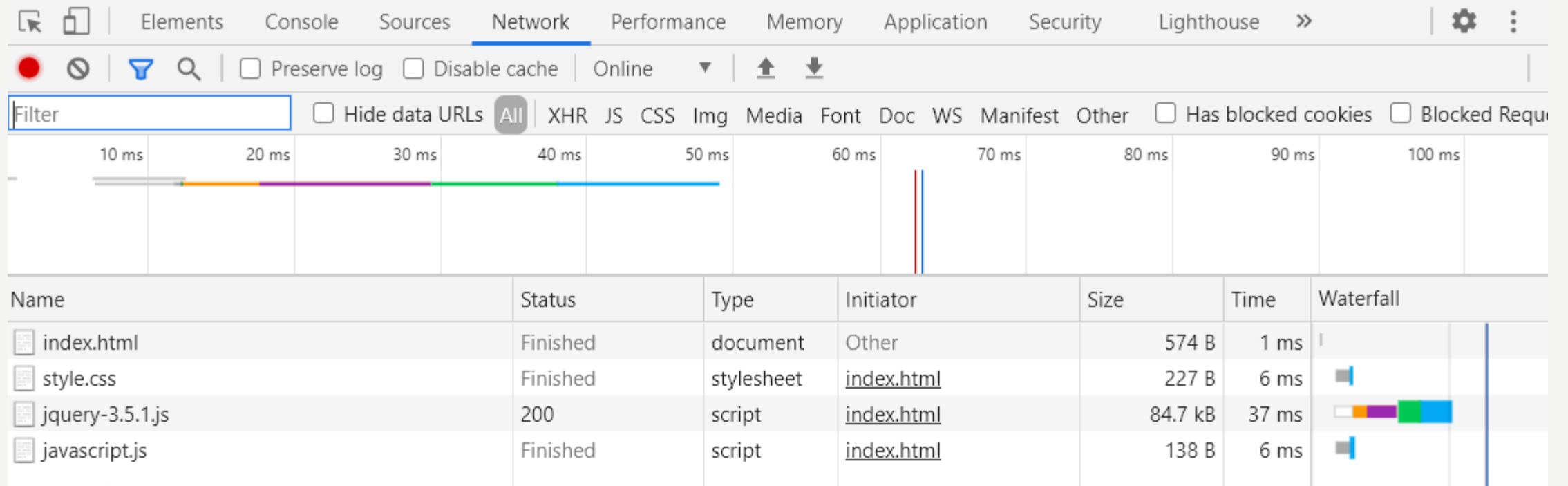


EXEMPLE SUR PERDU.COM



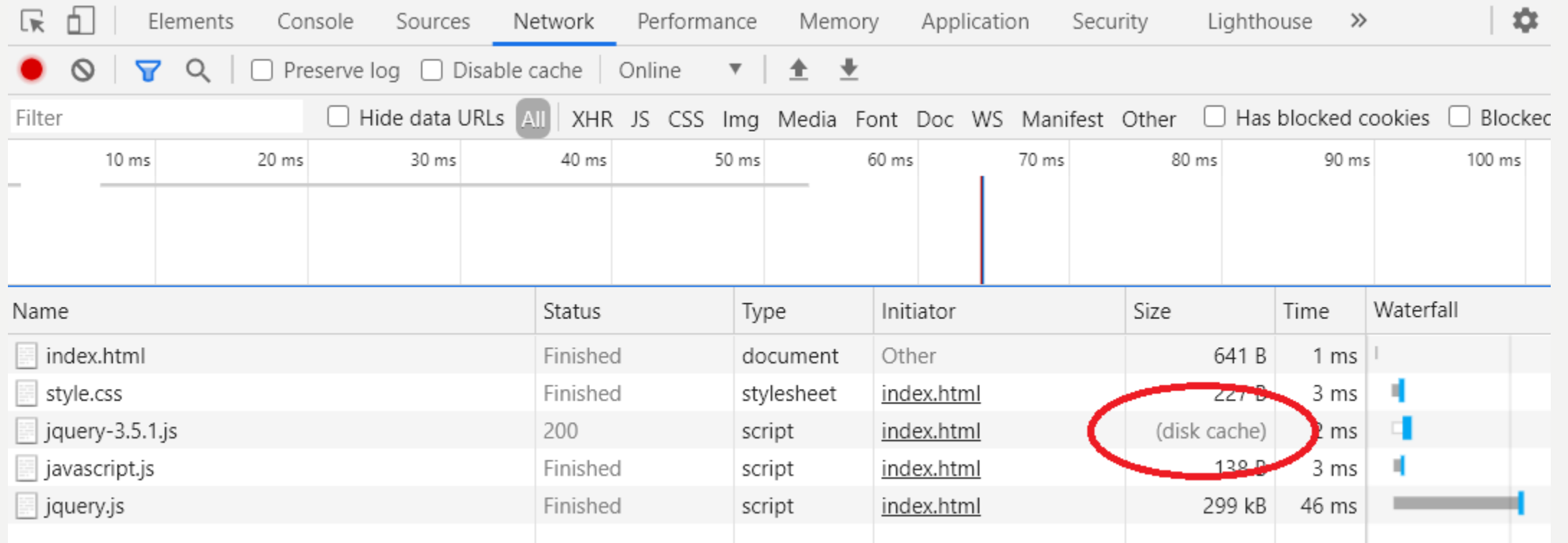
CDN : PREMIÈRE CONNEXION

- La récupération du fichier JQuery sur le CDN prends du temps et entraîne une connexion
- Si le CDN tombe, cela remet en cause l'intégrité de votre site



CDN : DEUXIÈME ACCES

- Sur la figure on aperçoit la mise en cache (GAIN DE TEMPS)
- La connexion n'a pas besoin de se refaire sauf si je vide le cache
- Une connexion a un coût énergétique, elle ne se produit que si le fichier n'est pas en cache. Le CDN est donc bénéfique à long terme (partagé par de nombreux sites)



W3TECHS : CHOISIR SON CDN

- https://w3techs.com/blog/entry/web_technologies_of_the_year_2019
- Ctrl+F → Content Delivery Network

JavaScript Content Delivery Network of the Year 2019

CDNJS

[CDNJS](#) is the fastest growing JavaScript content delivery network for the 4th time in a row. It is now used by more than 10% of all websites. [jsDelivr](#) and [unpkg](#) come in second and third, just like last year.

Result 2019

1. CDNJS
2. jsDelivr
3. unpkg

Winners of previous years

2019	CDNJS
2018	CDNJS
2017	CDNJS
2016	CDNJS
2015	Google Hosted Libraries
2014	jQuery CDN
2013	jQuery CDN

W3TECHS : CHOISIR UNE LIBRAIRIE

- https://w3techs.com/blog/entry/web_technologies_of_the_year_2019
- Ctrl+F → JavaScript Library

JavaScript Library of the Year 2019

Bootstrap

[Bootstrap](#) is winning the [JavaScript](#) library category for the 4th year in a row. [Popper](#) is now used by 2% of all websites, more that doubling its user base in 2019. [jQuery](#) is the most used [JavaScript](#) library by far and its usage is still growing, albeit slower than in previous years.

Result 2019

1. Bootstrap
2. Popper
3. jQuery

Winners of previous years

2019 Bootstrap
2018 Bootstrap
2017 Bootstrap
2016 Bootstrap
2015 jQuery
2014 jQuery
2013 jQuery
2012 jQuery
2011 jQuery

CHAPITRE 2 : LE DOM

RAPPEL DES OPÉRATIONS DANS LE DOM

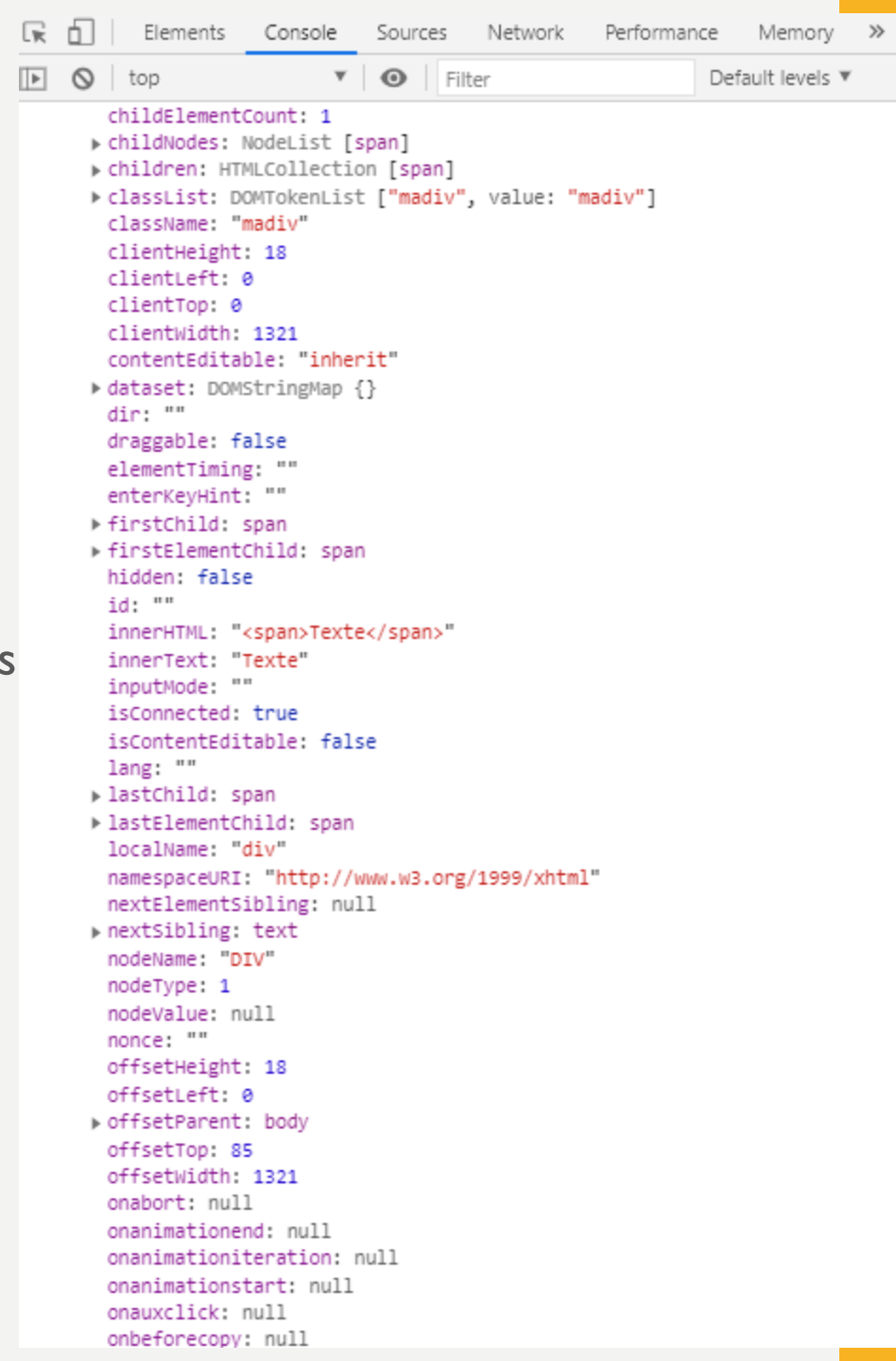
- Sélection d'éléments (Accès)
- Modification d'attributs
- Suppression d'attributs
- Ajout d'attributs
- Suppression d'éléments
- Ajout d'éléments
- Navigation
- Avec JQuery on ajoute le **déplacement**

A decorative wavy line in yellow and white, resembling a stylized lightning bolt or a path, runs vertically along the left side of the image.

ACCÈS AU DOM

LA SÉLECTION

- Grâce aux sélecteurs css3
 - `$("SELECTEUR CSS3");`
- L'objet obtenu possède environ 300 propriétés et méthodes
 - Elles permettent d'accéder ou de modifier l'apparence, les événements, leur position dans le dom, etc.



LA SÉLECTION

- Certaines de ces méthodes permettent d'affiner la sélection

- has()


```
<div class="madiv"><span>Texte</span></div>
```



```
<div class="class">Texte</div>
```

 →

```
$("#div").has("span").css("color","red");
```

 → 
- On sélectionne les div (`$("#div")`), puis, parmi les div, ceux qui ont un span (`.has("span")`) puis on change la couleur du/des élément(s) restant(s).
- not() → qui n'est pas du sélecteur en paramètre `$("#div").not(".class")` → retourne

```
<div class="madiv">
```
- first()/last() → le premier ou le dernier éléments de l'ensemble des éléments sélectionnés
- eq() → en paramètre on donne l'indice de l'élément désiré dans l'ensemble (indice commence à 0)
- slice() → découpe l'ensemble d'éléments en fonction des indices en paramètre

FILTER()

```
<div class="mdiv">
  <span>Texte1</span>
  <div class="c">Texte2</div>
</div>
```

```
2 $(document).ready(function(){
3
4   console.log($(".div.c").text());
5   console.log($(".div").filter(".c").text());
6
7 });
```

- Trouver l'utilité de filter()

<code>.filter()</code>	Filtre les éléments qui correspondent au sélecteur indiqué
------------------------	--

Elements	Console	»	⚙	⋮	×
🔍	🚫	top	▼	👁	Filter Defa ⚙
Texte2					javascript.js:4
Texte2					javascript.js:5
>					

jQuery filter() Method

The `filter()` method lets you specify a criteria. Elements that do not match the criteria are removed from the selection, and those that match will be returned.

The following example returns all `<p>` elements with class name "intro":

Definition and Usage

The `filter()` method returns elements that match a certain criteria.

This method lets you specify a criteria. Elements that do not match the criteria are removed from the selection, and those that match will be returned.

This method is often used to narrow down the search for an element in a group of selected elements.

Tip: The `filter()` method is the opposite of the `not()` method.

LES ÉLÉMENTS : GETTERS ET SETTERS

- Getters → Fonction qui retourne la valeur d'une propriété
- Setters → Fonction qui change ou affecte une valeur à une propriété
- En JQuery le getter et le setter sont la même fonction avec un nombre différent de paramètres
- Sur un ensemble d'éléments
 - Le getter retourne la valeur du premier éléments de l'ensemble
 - Le setter s'applique à l'ensemble des éléments

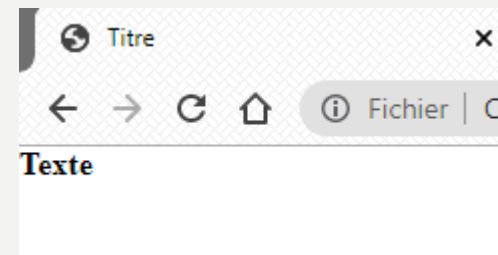
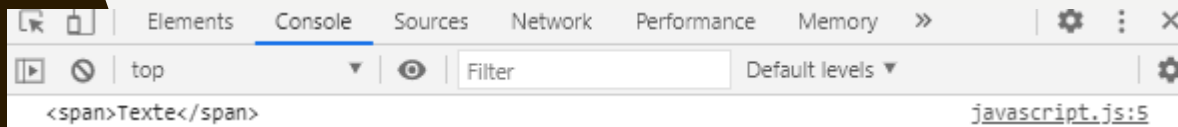
LES ÉLÉMENTS : GETTERS ET SETTERS

```
2  $(document).ready(function() {  
3  
4  //Getter  
5  console.log($("#madiw").html());  
6  
7  //Setter  
8  $("#madiw").html("<b>Texte</b>");  
9  
10 }
```

Ligne 5

Ligne 8

```
<body>  
  <div id="madiw"><span>Texte</span></div>  
</body>
```



LES ÉLÉMENTS : GETTERS ET SETTERS

LES PRINCIPAUX

- `html()` → retourne ou définit la descendance d'un élément (texte compris)
- `text()` → retourne ou définit le texte d'un élément (sans les balises html)
- `attr()` → définit ou retourne un attribut
- `css()` → définit ou retourne une propriété css

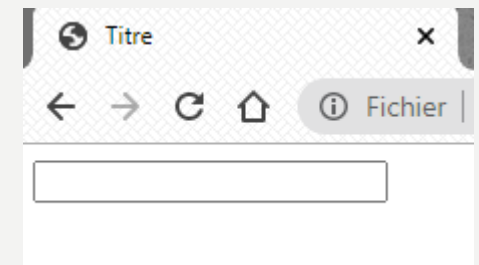
LES ÉLÉMENTS : GETTERS ET SETTERS HTML()

```
<div class="madiw">
  <span>Texte1</span>
  <div class="c">Texte2</div>
</div>
```

```
4 console.log($(".madiw").html());
```

```
<div class="madiw">
  <span>Texte1</span>
  <div class="c">Texte2</div>
</div>
```

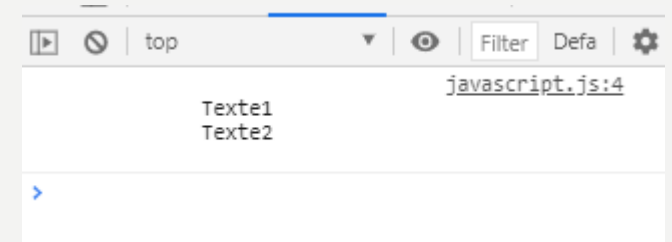
```
javascript.js:4
  <span>Texte1</span>
  <div class="c">Texte2</div>
4  $(".madiw").html("<input type='text' />");
```



LES ÉLÉMENTS : GETTERS ET SETTERS TEXT()

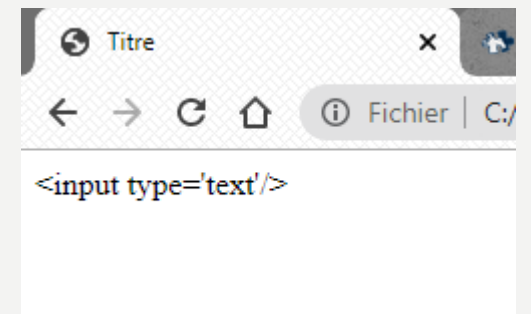
```
<body>  
  <div class="mdiv">  
    <span>Texte1</span>  
    <div class="c">Texte2</div>  
  </div>  
</body>
```

```
4 console.log($(".mdiv").text());  
5
```



```
<body>  
  <div class="mdiv">  
    <span>Texte1</span>  
    <div class="c">Texte2</div>  
  </div>  
</body>
```

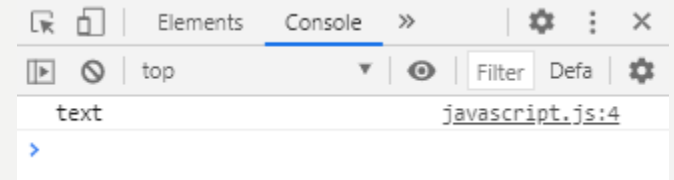
```
6 $(".mdiv").text("<input type='text'/>");  
7
```



LES ÉLÉMENTS : GETTERS ET SETTERS ATTR()

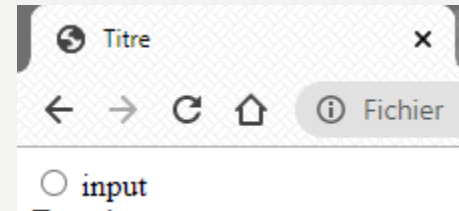
```
<input type="text" class="moninput"/> input
```

```
4 console.log($(".moninput").attr("type"));
```



```
<input type="text" class="moninput"/> input
```

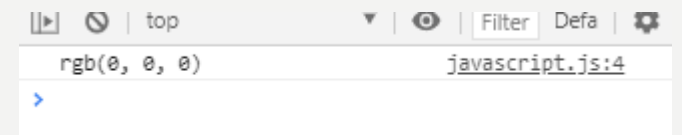
```
6 $(".moninput").attr("type", "radio");
```



LES ÉLÉMENTS : GETTERS ET SETTERS CSS()

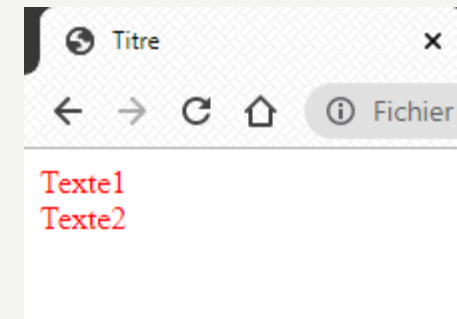
```
<div class="mdiv">
  <span>Textel</span>
  <div class="c">Texte2</div>
</div>
```

```
4 console.log($(".moninput").css("color"));
```



```
<div class="mdiv">
  <span>Textel</span>
  <div class="c">Texte2</div>
</div>
```

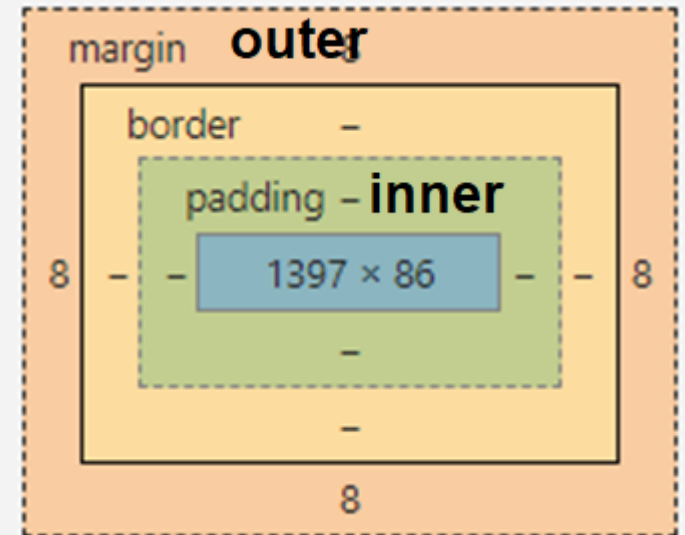
```
6 $(".mdiv").css("color", "red");
```



LES ÉLÉMENTS : GETTERS ET SETTERS

LA TAILLE

- `height()` → la hauteur de l'élément
- `innerHeight()` → la hauteur de l'élément avec le padding
- `outerHeight()` → la hauteur de l'élément avec le margin
- `width()` → la largeur de l'élément
- `innerWidth()` → la largeur de l'élément avec le padding
- `outerWidth()` → la largeur de l'élément avec le margin



LES ÉLÉMENTS : GETTERS ET SETTERS

LES AUTRES

- `offset()` → la position dans le document
- `scrollLeft()` → retourne ou définit la position de l'ascenseur horizontal
- `scrollTop()` → retourne ou définit la position de l'ascenseur vertical
- `position()` → la position par rapport au parent
- `val()` → retourne ou définit la valeur contenu dans l'attribut *value*

EACH : FONCTION POUR CHAQUE

- Le fonction each prend une fonction en paramètre et est appelée sur chaque élément d'un ensemble d'éléments

```
6 $( "div" ).each( function() {  
7     $( this ).css( "backgroundColor", "red" );  
8 } );  
9
```

- Va mettre la couleur de fond de toutes les divs en rouge

```
5  
6 $( "div" ).css( "backgroundColor", "red" );  
7
```

- On va utiliser pour faire des traitements plus compliqués, comme comparer des objets entre eux etc.

FILTER()

```
<div class="madiw">
  <span>Texte1</span>
  <div class="c">Texte2</div>
</div>
```

```
2 $(document).ready(function() {
3
4   console.log($(".div .c").text());
5   console.log($(".div").filter(".c").text());
6
7 });
```

- Trouver l'utilité de filter()

<code>.filter()</code>	Filtre les éléments qui correspondent au sélecteur indiqué
------------------------	--

Elements	Console	»	⚙	⋮	×
🔍	🚫	top	▼	👁	Filter Defa ⚙
Texte2					javascript.js:4
Texte2					javascript.js:5
>					

jQuery filter() Method

The `filter()` method lets you specify a criteria. Elements that do not match the criteria are removed from the selection, and those that match will be returned.

The following example returns all `<p>` elements with class name "intro":

Definition and Usage

The `filter()` method returns elements that match a certain criteria.

This method lets you specify a criteria. Elements that do not match the criteria are removed from the selection, and those that match will be returned.

This method is often used to narrow down the search for an element in a group of selected elements.

Tip: The `filter()` method is the opposite of the `not()` method.

A decorative wavy line in yellow and white on the left side of the image.

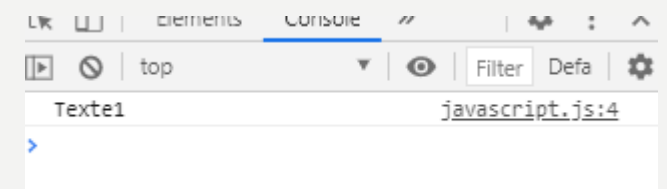
NAVIGATION DANS LE DOM

PARAMÈTRE OPTIONNEL

- `function test([selector])`
 - Signifie que la fonction possède un paramètre optionnel
- Toutes les fonctions de navigation peuvent prendre un sélecteur en paramètre qui appliquera un filtre supplémentaire

```
<div class="mdiv">  
  <span>Texte1</span>  
  <div class="c">Texte2</div>  
</div>
```

```
2  $(document).ready(function() {  
3  
4    console.log($(".div").children("span").text());  
5  
6  });  
7
```




LA NAVIGATION : PARENT

- `parent()` → Retourne le parent
- `parents()` → Retourne les aïeux
- `parentsuntil()` → Retourne les aïeux jusqu'à une condition d'arrêt
- `closest()` → Retourne le premier aïeul du type passé en paramètre
- ```
4 $("div").closest("div");
```

 → Retournera le premier aïeul de type div
- `offsetparent()` → Retourne le premier aïeul qui possède une propriété de positionnement (*position* en css)

# LA NAVIGATION : ENFANT

- `children()` → tous les enfants
- `find(selector)` → tous les éléments dans la descendance qui correspondent au selector
-  `.find( selector )` → absence de crochet, le paramètre n'est pas optionnel
- `contents()` → tous les enfants, ainsi que les nœuds de texte

# LA NAVIGATION : FRATRIE

- `siblings()` → tous les frères
- `next()` → le frère suivant
- `nextall()` → tous les frères suivants
- `nextuntil()` → tous les frères jusqu'à l'indice passé en paramètre
- `prev()` → le frère précédent
- `prevall()` → tous les frères précédents
- `prevuntil(x)` → tous les frères précédents jusqu'à l'indice passé en paramètre

A decorative wavy line in yellow and white on the left side of the image.

# **MODIFICATION DU DOM**



# CRÉER UN ÉLÉMENT

- 1ere méthode
  - Insertion dans le dom + l'html en texte

```
<body>
 <div class="mdiv">
 <div class="c">
 </div>
 </div>
</body>
```

```
4 $('div').append('<div id="newdiv">Test</div>');
5
```



```
<!DOCTYPE html>
<html>
 <head>...</head>
 <body>
 ... <div class="mdiv"> == $0
 <div class="c">
 <div id="newdiv">Test</div>
 </div>
 </body>
 </html>
```

# CRÉER UN ÉLÉMENT

- 2eme méthode
  - On crée l'objet
  - Puis on l'insère

```
13 <body>
14 <div class="madiw">
15 <div class="c">
16 </div>
17 </div>
18 </body>
19
```

```
3
4 var newObject = $('<div>');
5 newObject.attr("id", "newdiv");
6 $('#div').append(newObject);
7
```

Elements Console Sources

```
<!DOCTYPE html>
<html>
 <head>...</head>
 <body>
 <div class="madiw"> == $0
 <div class="c">
 <div id="newdiv"></div>
 </div>
 <div id="newdiv"></div>
 </div>
 </body>
</html>
```

# MANIPULATION DU DOM

- <https://api.jquery.com/category/manipulation/>
- Ajouter un élément avant/après un autre dans la fratrie : `insertAfter()`, `insertBefore()`, `after()`, `before()`
- Ajouter un élément au début/à la fin d'une descendance directe : `prepend()`, `prependTo()`, `append`, `appendTo()`
- Cloner un élément : `clone()`
- Remplacer un élément : `replace()`, `replaceAll()`
- Supprimer un élément : `remove()`, `detach()`
- Supprimer la descendance: `empty()`
- Capsule : `wrap()`, `unwrap()`, `wrapAll()`, `wrapInner()`

# ATTRIBUTS

- `removeAttr()` → supprimer un attribut
- `removeProp()` → supprimer une propriété
- `removeClass()` → supprimer un ou des classes
- `addClass()` → ajoute une ou des classes
- `toggleClass()` → permute une classe (si présent alors remove, sinon add)

# **CHAPITRE 2 : ÉVÉNEMENTS**

# RAPPEL

- On peut distinguer deux catégories d'événements
  - Action utilisateur
    - Souris
    - Clavier
  - Changement d'état d'un élément
- Grâce au JavaScript (et à plus petite échelle grâce au CSS) ces événements peuvent engendrer d'autres changements
  - Un événement peut donc déclencher d'autres événements
- La fonction déclenchée par un événement est appelée callback
- La liste des événements : <https://api.jquery.com/category/events/>

# CALLBACK : 1ERE MÉTHODE

- La fonction on()
  - Deux paramètres
    - L'événement
    - La fonction de callback

```
10 $("div").on('click', function() {
11 alert("click sur une div");
12 });
```

- Multiples événements

```
6 $("div").on({
7 click: function() {
8 alert("click sur une div");
9 },
10 mouseover: function() {
11 console.log($(this).offset());
12 }
13 });
14
```

- Particularité : un élément créé dynamiquement bénéficiera du callback

# CALLBACK : 2EME MÉTHODE

- Les fonctions événements
  - Prennent en paramètre une fonction de callback

```
4 $("div").click(function() {
5 alert("click sur une div");
6 });
7
```

- Particularité : ne s'applique pas aux éléments créés dynamiquement même s'ils correspondent au sélecteur



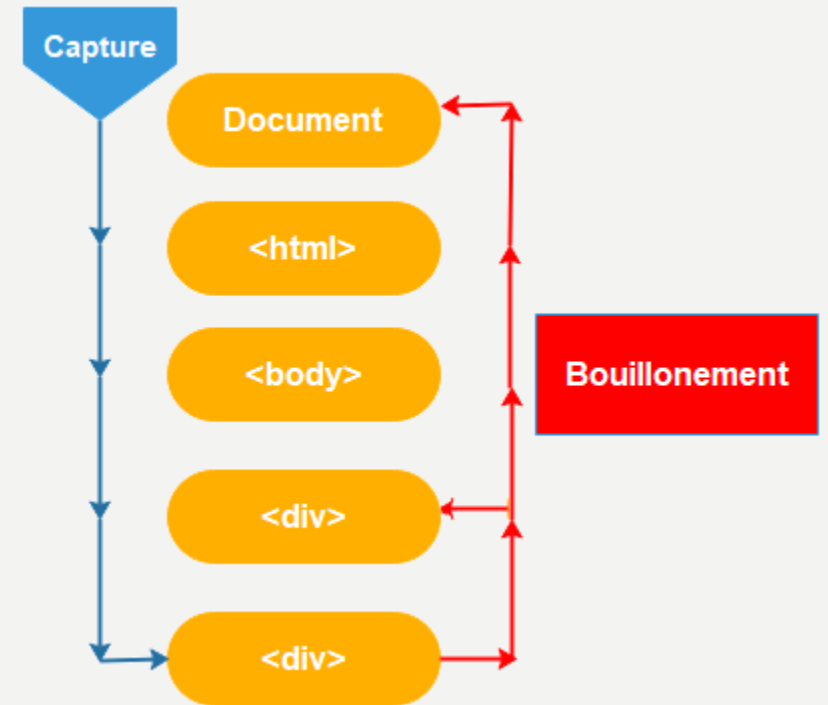
# CALLBACK : PARAMÈTRE

- Un callback peut prendre en paramètre l'événement
  - Accès aux détails de l'événement
- Dans l'exemple ci-dessous j'affiche le code de la touche pressée

```
5 $(document).keydown(function(e) {
6 console.log(e.keyCode);
7 });
8
9
```

# LA PROPAGATION D'ÉVÉNEMENT

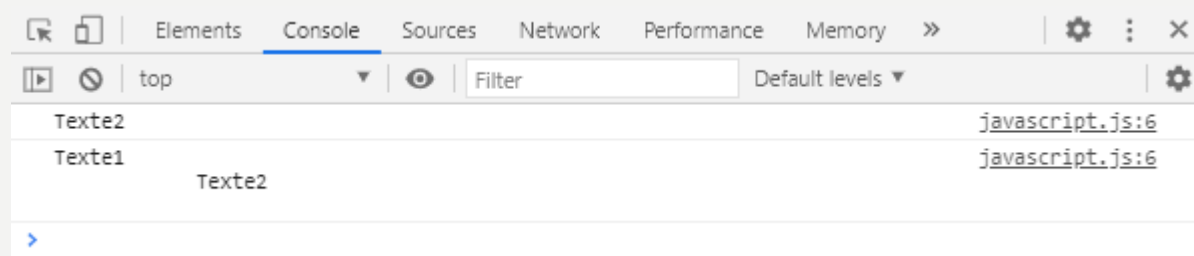
- Capture → élément ciblé → bouillonnement
- L'événement se propage jusqu'à l'élément déclencheur
- Le bouillonnement va re-déclencher l'événement si élément dans la parenté remplit les conditions
- Vous pourriez rencontrer le terme "dispatcher"  
c'est l'équivalent du bouillonnement  
Quand je clique sur div, je clique aussi sur tous les éléments dans lesquels il est contenu



# LA PROPAGATION D'ÉVÉNEMENT

```
13 <body>
14 <div class="madiv">
15 Texte1
16 <div class="c">Texte2</div>
17 </div>
18
19 </body>
```

```
5 $("div").click(function() {
6 console.log($(this).text());
7 });
```



# L'ARRÊT DU BOUILLONNEMENT

- Deux méthodes
  - `event.stopPropagation();`
  - `return false;`

```
5 = $("div").click(function(){
6 console.log($(this).text());
7 event.stopPropagation()
8
9 });
10 }
```

```
5 = $("div").click(function(){
6 console.log($(this).text());
7 return false;
8
9 });
10 }
```

# SUPPRIMER UN CALLBACK

- Supprimer tous les callback sur un événement
- Supprimer un des callback
  - Nécessite de ne pas utiliser de fonction anonyme
- Si je clique sur la seconde div :

```
6 $("div").off("click");
7
```

```
<div class="madiv">
 Texte1
 <div class="c">Texte2</div>
</div>
```

```
4 var i = 0;
5 var mafonction = function() {
6 console.log($(this).text());
7 }
8 var mafonction2 = function() {
9 console.log(i);
10 i++;
11 }
12 $("div").on("click", mafonction);
13 $("div").on("click", mafonction2);
14 $("div").off("click", mafonction);
15
```



Elements Console >>		Filter Defa	Settings	Close
top				
0	javascript.js:9			
1	javascript.js:9			
>				

# DÉCLENCHER UN ÉVÉNEMENT

- La fonction *trigger* va appeler tous les callbacks de l'événement passé en paramètre

```
<div class="mdiv">
 Texte1
 <div class="c">Texte2</div>
</div>
```

```
4 var i = 0;
5 var mafonction2 = function() {
6 console.log(i);
7 i++;
8 }
9 $("div").on("click", mafonction2);
10 $("div").trigger("click");
11
```



The screenshot shows a web browser's developer console with the 'Console' tab selected. It displays three log entries, each showing the value of 'i' (0, 1, and 2) and the file 'javascript.js:6'. The console also shows the prompt character '>' at the bottom.

0	javascript.js:6
1	javascript.js:6
2	javascript.js:6
>	



# **CHAPITRE 3 :**

# **EFFETS/ANIMATIONS**

# POURQUOI UTILISER JQUERY PLUTÔT QUE CSS3 ?

- Des fonctions toutes faites accélérant le développement
- Une rétro compatibilité avec les navigateurs
- La fluidité
- La possibilité d'y associer un callback



# LES FONCTIONS TOUTES FAITES

- `toggle()` → si caché alors montrer, sinon cacher
- `fadeIn()` → apparition par fondu
- `fadeOut()` → disparition par fondu
- `fadeToggle()` → si visible alors disparition par fondu, sinon apparition par fondu
- `fadeTo()` → apparition par fondu jusqu'à un certain niveau d'opacité
- `slideDown()` → apparition en glissé vers le bas
- `slideUp()` → disparition en glissé vers le haut
- `slideToggle()` → si visible alors disparition en glissé vers le haut, sinon apparition en glissé vers le bas

# LES FONCTIONS TOUTES FAITES

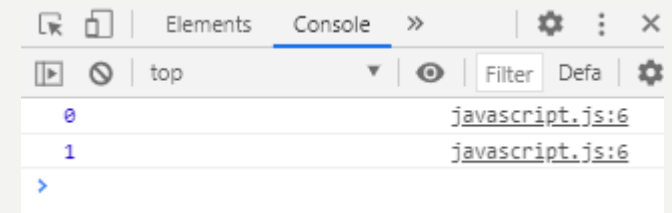
- Elles possèdent 3 paramètres optionnels, dans l'ordre :
  - Un paramètre qui précise la durée de l'animation en ms
    - Ce paramètre vaut 400 par défaut
  - Un paramètre qui précise la fonction d'accélération
  - Un paramètre de callback
- `fadeTo()` prend un quatrième paramètre, qui se place en deuxième position, pour déterminer l'opacité visée entre 0 et 1

# LES FONCTIONS TOUTES FAITES : EXEMPLE

- 4000 → l'animation dure 4s
- mafonction2 → callback appelé à la fin de l'animation

```
<div class="madiv">
 Texte1
 <div class="c">Texte2</div>
</div>
```

```
3
4 var i = 0;
5 var mafonction2 = function(){
6 console.log(i);
7 i++;
8 }
9 $("div").fadeOut(4000, mafonction2);
10
```



# LES FONCTIONS TOUTES FAITES

- `hide()` → cacher → `fadeOut` + `slideUp`
- `show()` → montrer → `fadeIn` + `slideDown`
- Prends également en paramètre la vitesse, l'accélération et le callback

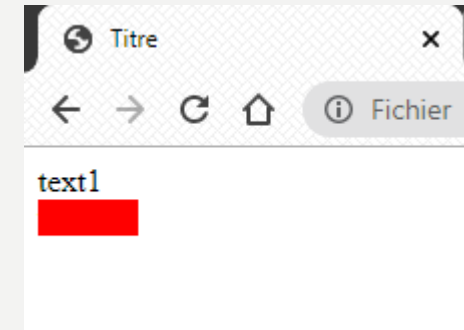
```
4 var i = 0;
5 var mafonction2 = function() {
6 console.log(i);
7 i++;
8 }
9 $("div").hide(4000, mafonction2);
```

# ANIMATION

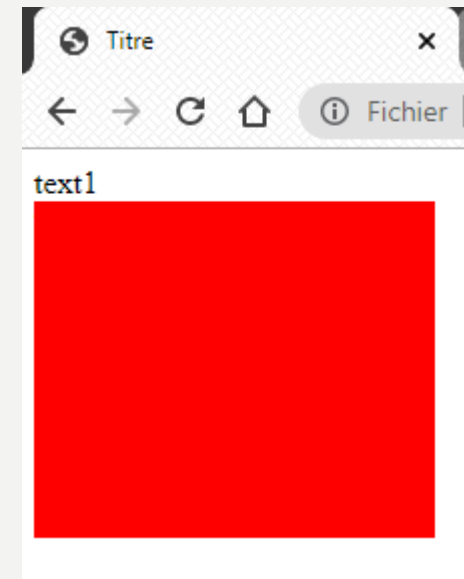
- animate(), prend en paramètre
  - Ensemble de paramètres
  - Durée de l'animation
  - Callback de fin d'animation

```
<div class="c" style="width:50px;
background-color:red">
</div>
```

```
4 $('.c').click(function() {
5 $(this).animate({
6 height: '+=150px',
7 width: '+=150px'
8 }, 1000);
9 });
```



click



# ANIMATION

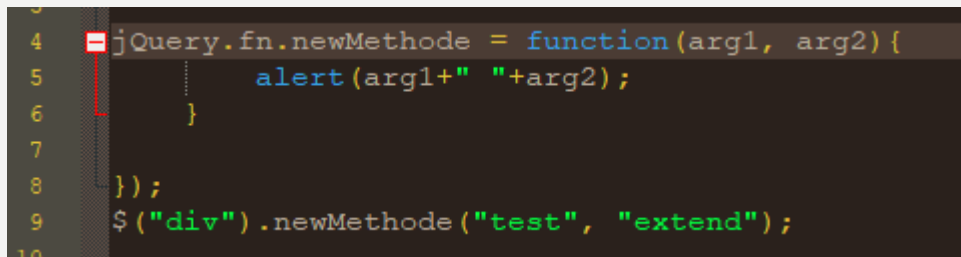
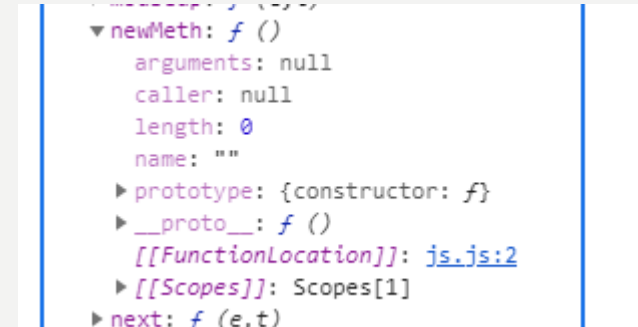
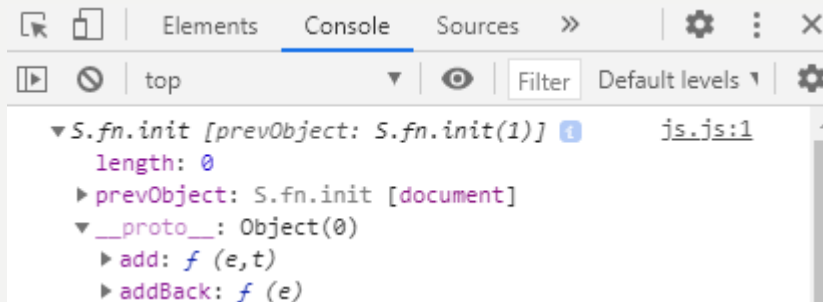
- Il est possible de chaîner les animations
- D'arrêter l'animation avec `stop()`
- De retarder le départ de l'animation avec `delay()`

```
4 = $('c').click(function(){
5 = $(this).animate({
6 height: '+=150px'
7 }, 1000)
8 .animate({
9 width: '+=150px'
10 }, 1000)
11 });
12
```

# **CHAPITRE 4 : ETENDRE JQUERY**

# CRÉER UNE MÉTHODE 1

- `jQuery.fn.maMeth = function()`
- `fn ?`



Cette page indique

test extend

OK



# CRÉER UNE MÉTHODE 2

- extend()
  - L'objet à étendre
  - La propriété à étendre
  - La nouvelle propriété/fonction

```
4 $.extend(jQuery.fn, {
5
6 newMethode : function(arg1, arg2){
7 alert(arg1+" "+arg2);
8 }
9
10 });
11 $("div").newMethode("test", "extend");
12
```

Cette page indique  
test extend

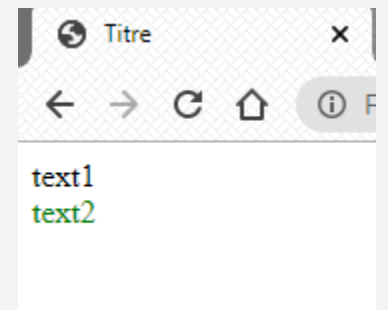
OK

# CRÉER UN SÉLECTEUR

- Étendre `jQuery.expr[':']`
- Donner un nouveau sélecteur et y associer une fonction qui retourne vrai si l'élément doit être sélectionné

```
</head>
<body>
 <div class="madiv">
 text1
 </div>
 <div class="c" style="color:red">
 text2
 </div>
</body>
```

```
4 = $.extend(jQuery.expr[':'], {
5
6 couleur : function(a, i, m){
7 return $(a).css('color') == m[3];
8 }
9
10 });
11
12 $("div:couleur('rgb(255, 0, 0)')").css("color", "green");
```

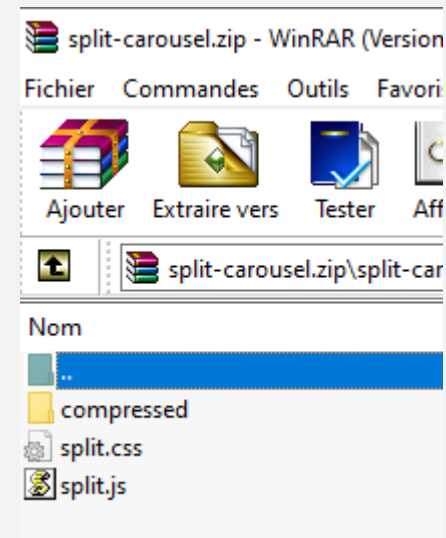


# PLUGIN

- Télécharger un plugin : <https://www.jqueryscript.net/jquery-plugins/>
- Split carroussel.zip
- Contient les instructions simples d'installations dans le fichier README
- Contient un ensemble de fichiers css et js (ici seulement 2, parfois plus, parfois d'autres types)

## How to setup

1. Include split.scss ( or split.min.css ) anywhere on the page  
`<link rel="stylesheet" type="text/css" href="split.css">`
2. Include split.js ( or split.min.js ) AFTER jQuery.js  
`<script type="text/javascript" src="split.js"></script>`



# VANILLA JS ?

- <https://www.blogduwebdesign.com/vanilla-js-le-framework-javascript-le-plus-utilise-au-monde/>
  - 2017 : léger, à la base de jquery, utilisé par google, fb, twitter, amazon...
- <http://vanilla-js.com/>

## Retrieve DOM element by ID

	Code	ops / sec
<i>Vanilla JS</i>	<code>document.getElementById('test-table');</code>	12,137,211
<b>Dojo</b>	<code>dojo.byId('test-table');</code>	5,443,343
<b>Prototype JS</b>	<code>\$('#test-table')</code>	2,940,734
<b>Ext JS</b>	<code>delete Ext.elCache['test-table']; Ext.get('test-table');</code>	997,562
<b>jQuery</b>	<code>\$jq('#test-table');</code>	350,557
<b>YUI</b>	<code>YAHOO.util.Dom.get('test-table');</code>	326,534
<b>MooTools</b>	<code>document.id('test-table');</code>	78,802

# VERS D'AUTRES LIBRAIRIES

- <https://www.artwai.com/faut-il-enterrer-jquery/>
  - 2019 : JQuery 3 et pourtant beaucoup de sites sont toujours sur JQuery l
- <https://www.ma-no.org/en/programming/javascript/is-jquery-going-to-die-in-2019>
  - 2019 : JQuery n'est pas mort, mais en déclin... A utiliser avec parcimonie
- Est-il nécessaire d'utiliser une librairie ?
  - Si oui pourquoi ?
  - Si oui laquelle ?
  - Si oui comment ?

# SITOGRAPHIE

- W3schools : <https://www.w3schools.com/jquery/default.asp>
- Documentation officiel : <https://api.jquery.com/>
- Site officiel jquery : <https://jquery.com/download/>
- Technologie web en vogue : <https://w3techs.com>