



Introduction et concepts de base

Principe de chaînage

- Schématiquement :

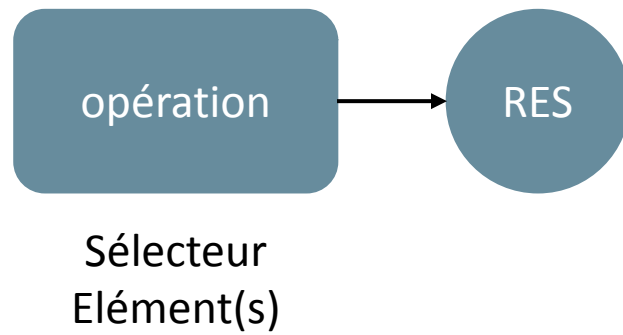


opération

Sélecteur
Élément(s)

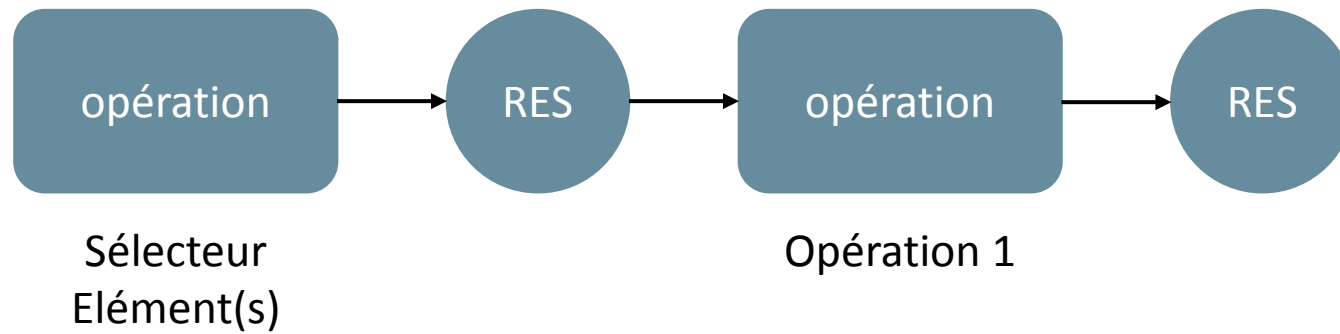
Principe de chaînage

- Schématiquement :



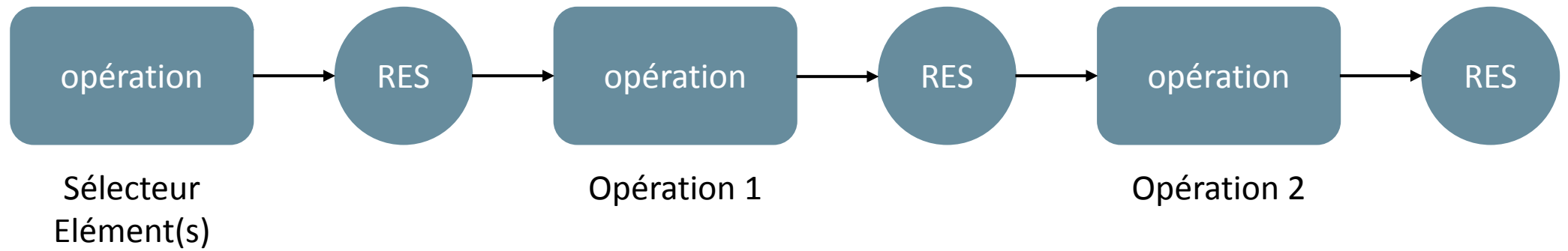
Principe de chaînage

- Schématiquement :



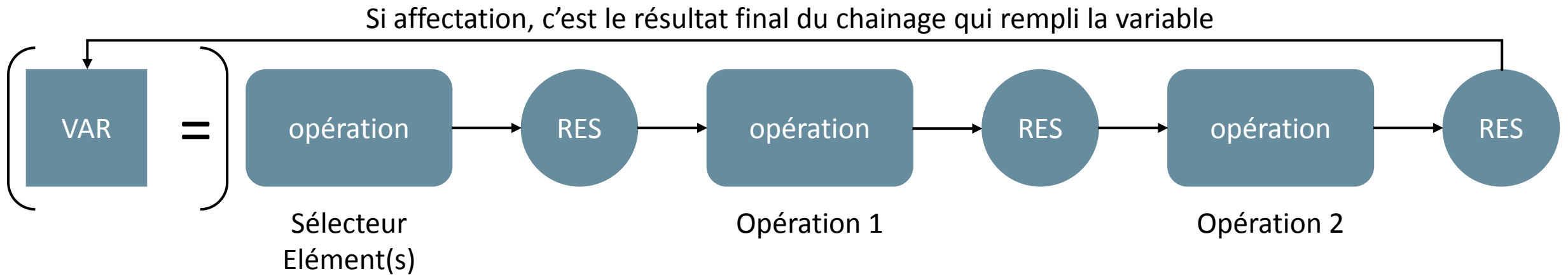
Principe de chaînage

- Schématiquement :



Principe de chaînage

- Schématiquement :



Principe de chaînage

- Plus court, plus léger, plus clair, plus mieux !
- Attention a prévoir les valeurs de retour.

```
//Sans Chainage  
$( "#maliste" ).removeClass( "active" );  
$( "#maliste" ).addClass( "hidden" );  
$( "#maliste" ).hide( );
```

Principe de chaînage

- Plus court, plus léger, plus clair, plus mieux !
- Attention a prévoir les valeurs de retour.

```
//Sans Chainage
```

```
$( "#maliste" ).removeClass( "active" );
```

```
$( "#maliste" ).addClass( "hidden" );
```

```
$( "#maliste" ).hide();
```

```
//Avec Chainage
```

```
$( "#maliste" ).removeClass( "active" ).addClass( "hidden" ).hide();
```


Principe de chaînage

- Plus court, plus léger, plus clair, plus mieux !
- Attention a prévoir les valeurs de retour.

```
//Sans Chainage
```

```
$( "#maliste" ).removeClass( "active" );  
$( "#maliste" ).addClass( "hidden" );  
$( "#maliste" ).hide( );
```

```
//Avec Chainage
```

```
$( "#maliste" ).removeClass( "active" ).addClass( "hidden" ).hide( );  
$( "#maliste" )  
    .removeClass( "active" )  
    .addClass( "hidden" )  
    .hide( );
```



Les Callbacks c'est du propre

```
function passerAspirateur(piece, callback) {  
    console.log('je passe l'aspirateur dans ' + piece + '.');  
    console.log('j'ai terminé.');
```



```
    if(callback) { callback(); }  
}  
  
function rangerAspirateur() {  
    console.log('Aspirateur rangé.');
```



```
}  
  
passerAspirateur('salon');  
passerAspirateur('cuisine', rangerAspirateur);
```



Les Callbacks c'est du propre

```
function passerAspirateur(piece, callback) {  
    console.log('je passe l'aspirateur dans ' + piece + '.');  
    console.log('j'ai terminé.');
```



```
    if(callback) { callback(); }  
}  
  
function rangerAspirateur() {  
    console.log('Aspirateur rangé.');
```



```
}  
  
passerAspirateur('cuisine', function() {  
    rangerAspirateur();  
    faireLesCourses();  
});
```



Les Callbacks c'est du propre

```
function passerAspirateur(piece, callback) {  
    console.log('je passe l'aspirateur dans ' + piece + '.');  
    console.log('j'ai terminé.');
```



```
    if(callback) { callback(); }  
}  
  
function rangerAspirateur() {  
    console.log('Aspirateur rangé.');
```



```
}  
  
passerAspirateur('cuisine', function() {  
    passerAspirateur('cuisine', rangerAspirateur);  
    faireLesCourses();  
});
```



Manipulation « graphique »

- Des méthodes prêtes à l'emploi :

AFFICHER / MASQUER

```
.hide(sp,cb)  
.show(sp,cb)  
.toggle(sp,cb)
```

SLIDE

```
.slideDown(sp,cb)  
.slideUp(sp,cb)  
.slideToggle(sp,cb)
```

FADE

```
.fadeIn(sp,cb)  
.fadeOut(sp,cb)  
.fadeToggle(sp,cb)
```

- Des paramètres identiques et optionnels
 - speed : vitesse de l'animation (slow | fast | x Millisecondes)
 - Callback : fonction de rappel, exécutée après la fin de l'animation



Manipulation « graphique »

```
function monRappel() {  
    console.log('animation terminée.');}  
  
//Affiche le span#to-open (400ms)  
$('span#to-open').show();  
  
//Affiche le span#to-open-2 rapidement (200ms)  
$('span#to-open-2').show("fast");  
  
//Cache le span#to-hide et exécute la fonction monRappel une fois terminé. (600ms)  
$('span#to-hide').hide("slow", monrappel);  
  
//Cache le span#to-hide et exécute la fonction anonyme une fois terminé. (600ms)  
$('span#to-hide').hide("slow", function() {  
    console.log('Rappel anonyme : animation terminée.');});
```



Gérer les évènements

- JavaScript : programmation événementielle
- Réagir aux actions des utilisateurs
- Réagir à des événements internes
- Événements préprogrammés (utilisateur principalement)
- Événements personnalisés possibles
- Les callbacks ici sont plutôt appelés handlers



Gérer les évènements

//Gère le clic utilisateur

```
$('#a#myLinkTarget').click(function() {  
    console.log('Vous avez cliquer sur le lien.');
```

//Gère l'envoi d'un formulaire

```
$('#form#add-user-form').submit(function() {  
    console.log('Vous avez envoyé le formulaire.');
```

//Gère le changement de valeur ou d'état d'un champ

```
$('#input#unstable').change(function() {  
    console.log('La valeur du champ a changé.');
```

```
});
```




Gérer les évènements

```
//Gère le survol
```

```
$('#a#myLinkTarget').hover(function() {  
    console.log('Vous êtes sur le lien.');
```



```
}, function() {  
    console.log('Vous n'êtes plus au dessus du lien.');
```



```
});
```

```
//Si l'évènement n'a pas de fonction prédéterminée
```

```
$('#input#you-type').bind('keydown', function() {  
    console.log('Vous avez pressé une touche.');
```



```
});
```

Les évènements à la carte

```
//Déclenchement d'un évènement personnalisé
```

```
function jeFaisUnePizza() {  
    console.log('J'ai terminé.');
```

```
    $('body').trigger('pizza_ready');
```

```
}
```

```
//Si l'évènement n'a pas de fonction prédéterminée
```

```
$('#body').bind('pizza_ready', function() {  
    $('#div.assiette').append('<span class="part-de-pizza">miam</span>');
```

```
    console.log('A table !');
```

```
});
```

```
jeFaisUnePizza();
```



Gérer les évènements

```
//Gère le clic utilisateur
```

```
$( 'a#myLinkTarget' ).click(function(event) {  
    event.preventDefault(); //Empêche le lien de faire son action standard  
    console.log('Vous avez cliquer sur le lien.');
```

```
});
```

jQuery et ses plugins

- Beaucoup, beaucoup (trop ?) de plugins
- Avant d'installer un plugin : en avez-vous besoin ?
 - Il arrive que quelques lignes sympas suffisent
- Attention à la qualité / stabilité du code
 - N'installez pas n'importe quoi sur un site en prod
- Date de mise à jour : si le plugin a 5 ans, est-il encore compatible
- Compatibilité avec la version de jQuery, c'est pas qu'une question de date
- Trop de plugin tue le plugin : ne pas charger 900Ko de JavaScript !
- Dépôt officiel : <https://plugins.jquery.com/>



Liens pour aller plus loin

<https://developer.mozilla.org/fr/docs/JavaScript>

[https://developer.mozilla.org/fr/docs/Web/JavaScript/Une r%C3%A9introduction %C3%A0 JavaScript](https://developer.mozilla.org/fr/docs/Web/JavaScript/Une_r%C3%A9introduction_%C3%A0_JavaScript)

<http://api.jquery.com/category/selectors/>

<http://www.caenjs.org/>

<http://fr.openclassrooms.com/informatique/cours/tout-sur-le-javascript>

<http://javascript.developpez.com/cours/>

<http://www.js-attitude.fr/>

<http://www.grafikart.fr/tutoriels/javascript>