

this, \$(this) et e.target

```
<ul id="monUL">
  <li>Un</li>
  <li>Deux</li>
  <li>Trois</li>
</ul>
```

```
$monUL = document.getElementById("monUL");
function analyser(e) {
  alert(this); // le UL qui a déclenché le onclick
  alert(e.target); // le LI qui a été cliqué
}
$monUL.onclick = analyser;
```

En jQuery, `$(this)` permet de transformer un élément JavaScript classique en élément jQuery, afin de pouvoir utiliser toutes les fonctions de jQuery dessus.

```
function analyser(e) {
  alert($(this)); // le UL en mode jQuery
  this.addClass(...) // impossible
  this.classList.add(...) // possible
  $(this).addClass(...) // possible
  $(this).classList.add(...) // impossible
}
$("#monUL").click(analyser);
```

Mais nous aurions directement pu faire :

```
$("#monUL").on("click", "li", analyser);
```

Ainsi, le `$(this)` donne directement le ``. Très pratique, car **cela continuera de fonctionner si de nouveaux `` apparaissent** dans le ``.

Équivalent du innerHTML en jQuery

```
$(element).html(); // récupère le contenu HTML de l'élément
$(element).html("..."); // remplace le contenu
$(element).html($(element).html() + "..."); // ajoute du contenu
```

Exercice : agenda (encore !)

Refaire l'agenda (affichage, ajout et suppression) sauce jQuery.

Agenda minimaliste, alternatif, avec des morceaux de jQuery

<https://jsfiddle.net/hmu2oeqz/>

Exercice : carousel jQuery

Étape 1 : extraire le JSFiddle dans des fichiers HTML / CSS / JS. Pensez à inclure jQuery avant votre JS à vous !

Étape 2 : inclure un plugin jCarousel, lire sa documentation, et faire un carousel à partir de la liste de contacts.

