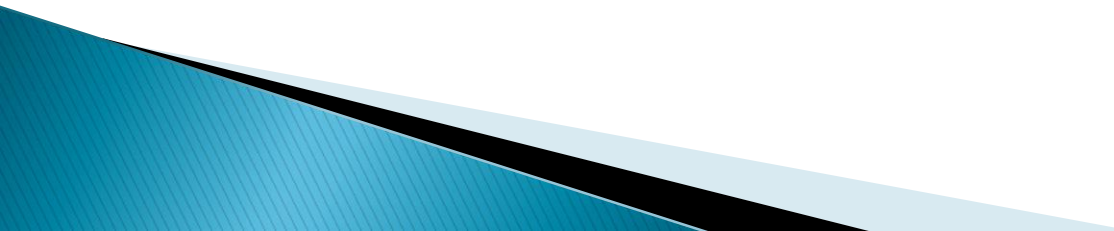


AJAX

Asynchronous JavaScript and XML

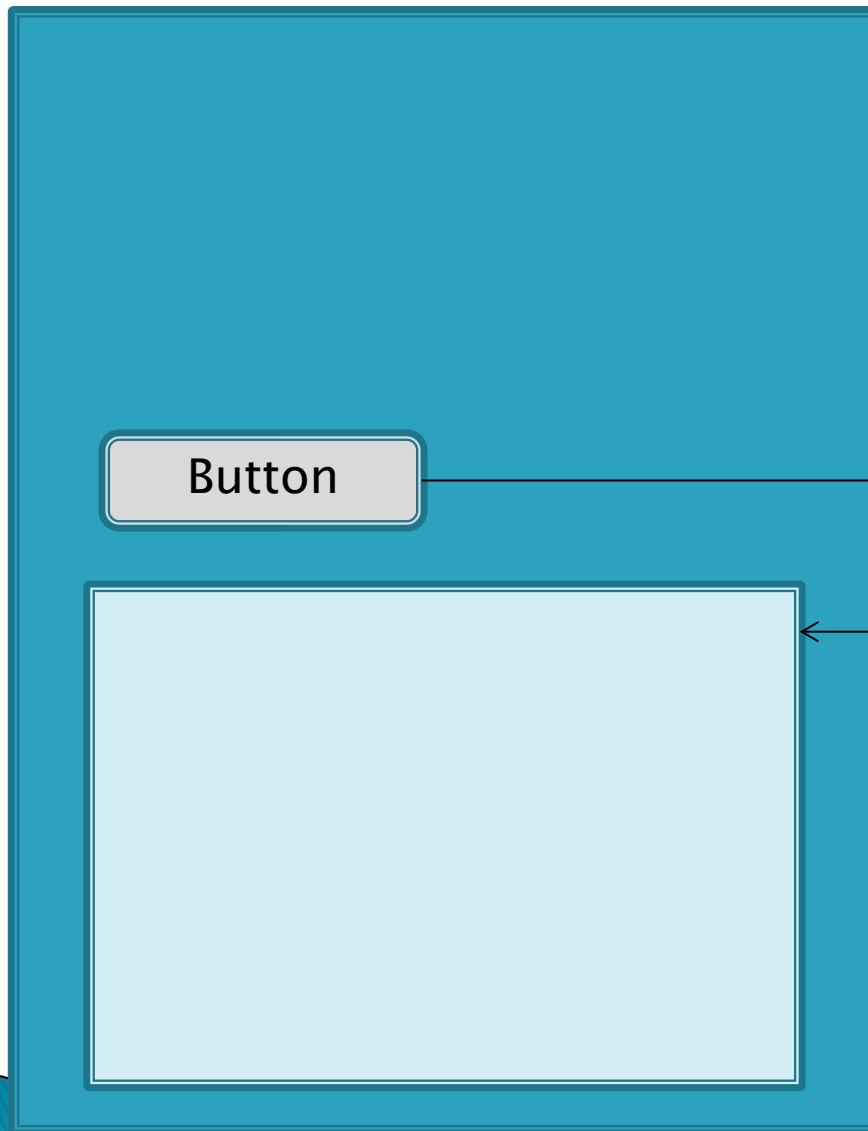
Wikipédia

L'architecture informatique Ajax (*Asynchronous JavaScript and XML*) permet de construire des applications Web et des sites web *dynamiques interactifs* sur le poste client en se servant de différentes technologies ajoutées aux navigateurs web entre 1995 et 2005.



A la base Ajax repose sur l'utilisation de l'objet Javascript XMLHttpRequest.

Cet objet permet de lancer une requête HTTP (sans recharger la page) et de placer le résultat dans un élément HTML



Première requête HTTP
Chargement de la page

Deuxième requête HTTP
Sans rechargement de la page
mise à jour du contenu
d'un élément HTML

AJAX avec JQuery

```
$("#div1").load("contenu.php");
```

Ce code permet de charger le contenu de **contenu.php** et de placer le résultat dans l'élément HTML qui port l'id **div1**

Attention: contenu.php ne doit pas être une page HTML complète



AJAX avec JQuery

```
$.get("contenu.php", null, function( data ) {  
    $("#div1").html(data);  
});
```

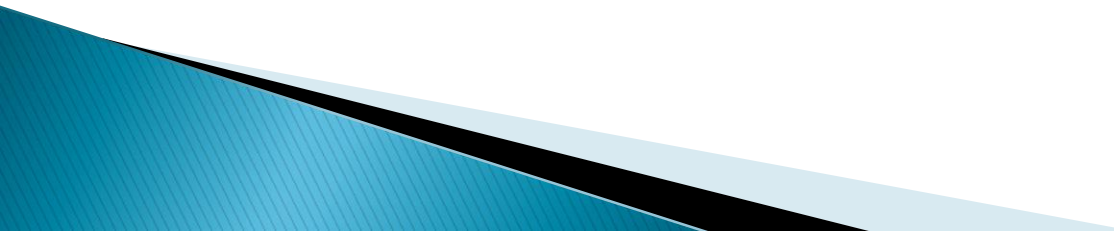
Deuxième façon d'utiliser AJAX avec la méthode **get**
Avec 3 paramètres:

1. url de la page à charger
2. données à envoyer (en GET)
3. fonction à déclencher en cas de succès

AJAX avec JQuery

```
$.ajax({  
    url: "contenu.php",  
    success: function( data ) {  
        $("#div1").html(data);  
    },  
});
```

La méthode **ajax** de JQuery permet de spécifier l'**url** de la page à charger et la fonction à exécuter en cas de **succès**.



Envoi de données 1

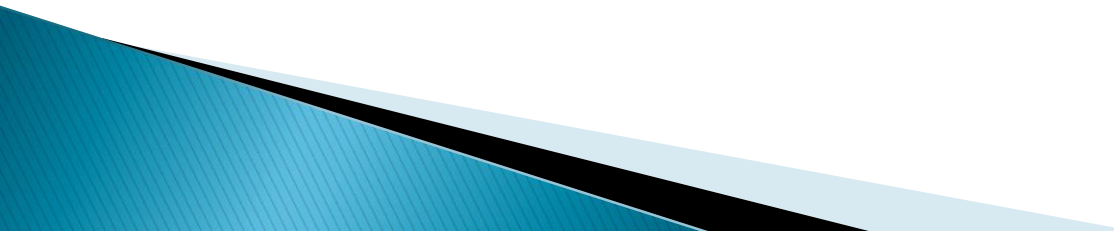
```
$.get("contenu.php", "id=27", function(  
data ) {  
    $("#div1").html(data);  
});
```

La méthode **get** permet d'envoyer des données avec le deuxième paramètre (dans l'exemple id=27)

Envoi de données 2

```
$.post("contenu.php", "id=27", function(  
data ) {  
    $("#div1").html(data);  
});
```

La méthode **post** permet d'envoyer des données avec le deuxième paramètre (dans l'exemple id=27)



Envoi de données 3

```
$.post(  
    "contenu.php", $("#form1").serialize(),  
    function( data ) {  
        $("#div1").html(data);  
    }  
);
```

La méthode `serialize` permet de contruire une chaine "name=value&name=value" qui représente les données du formulaire (id=form1)