



# AJAX & APIS

## HTTP REQUEST METHODS

---

GET	<i>Retrieve a resource</i>
POST	<i>Create a resource</i>
PATCH	<i>Update an existing resource</i>
PUT	<i>Replace an existing resource</i>
DELETE	<i>Delete a resource</i>

## HTTP STATUS CODES

---

200	<i>Okay</i>
301	<i>Moved permanently</i>
302	<i>Moved temporarily</i>
400	<i>Bad request</i>
403	<i>Forbidden</i>
404	<i>Not found</i>
500	<i>Internal server error</i>

## FETCH REQUEST

---

```
fetch(url)
  .then((response) => {
    // check if request was successful
  })
  .then((data) => {
    // do something with the response
  });
```

## JQUERY AJAX

---

<code>\$.get(url);</code>	<i>loads data from the url using an HTTP GET request</i>
---------------------------	--

<code>\$.ajax({...});</code>	<i>performs an Ajax request based on properties you specify in the object passed as a parameter</i>
------------------------------	---

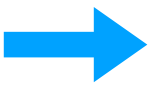
---

# JQUERY BEST PRACTICES

## CHAINING

---

```
$( '<p>' );  
$mainCaption.text( 'Today' );  
$captionWithText.addClass( 'accent' );
```



```
$( '<p>' ).text( 'Today' ).addClass( 'accent' );
```

## IMPLICIT ITERATION

---

```
$( 'li' ).each( function() {  
    $( this ).removeClass( 'current' );  
});
```





```
$( 'li' ).removeClass( 'current' );
```

## EVENT DELEGATION

---

Event listener set on  
parent element

Second argument added to on() method, specifying child elements to listen for events on



```
$listElement.on( 'mouseenter', 'li', function( event ) {  
    $( this ).siblings().removeClass( 'active' );  
    $( this ).addClass( 'active' );  
});
```

## ATTACHING MULTIPLE EVENTS WITH A SINGLE .ON() STATEMENT

---

```
$listElement.on( 'mouseenter mouseleave', 'li', function( event ) {  
    if ( event.type === 'mouseenter' ) {  
        $( this ).siblings().removeClass( 'active' );  
        $( this ).addClass( 'active' );  
    } else if ( event.type === 'mouseleave' ) {  
        $( this ).removeClass( 'active' );  
    }  
});
```