AJAX & APIS

HTTP REQUEST METHODS

`GET`	*Retrieve a resource*
`POST`	*Create a resource*
`PATCH`	*Update an existing resource*
`PUT`	*Replace an existing resource*
`DELETE`	*Delete a resource*

HTTP STATUS CODES

`200`	*Okay*
`301`	*Moved permanently*
`302`	*Moved temporarily*
`400`	*Bad request*
`403`	*Forbidden*
`404`	*Not found*
`500`	*Internal server error*

FETCH REQUEST

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

JQUERY AJAX

`$.get(url);`	*loads data from the url using an HTTP GET request*
`$.ajax({…});`	*performs an Ajax request based on properties you specify in the object passed as a parameter*

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');
  }
});
```