



AJAX

ASYNCHRONOUS JS

OVERVIEW

- Intro to asynchronous code
- xmlhttp Request
- jQuery Ajax
- Fetch
- Async Await
- Axios

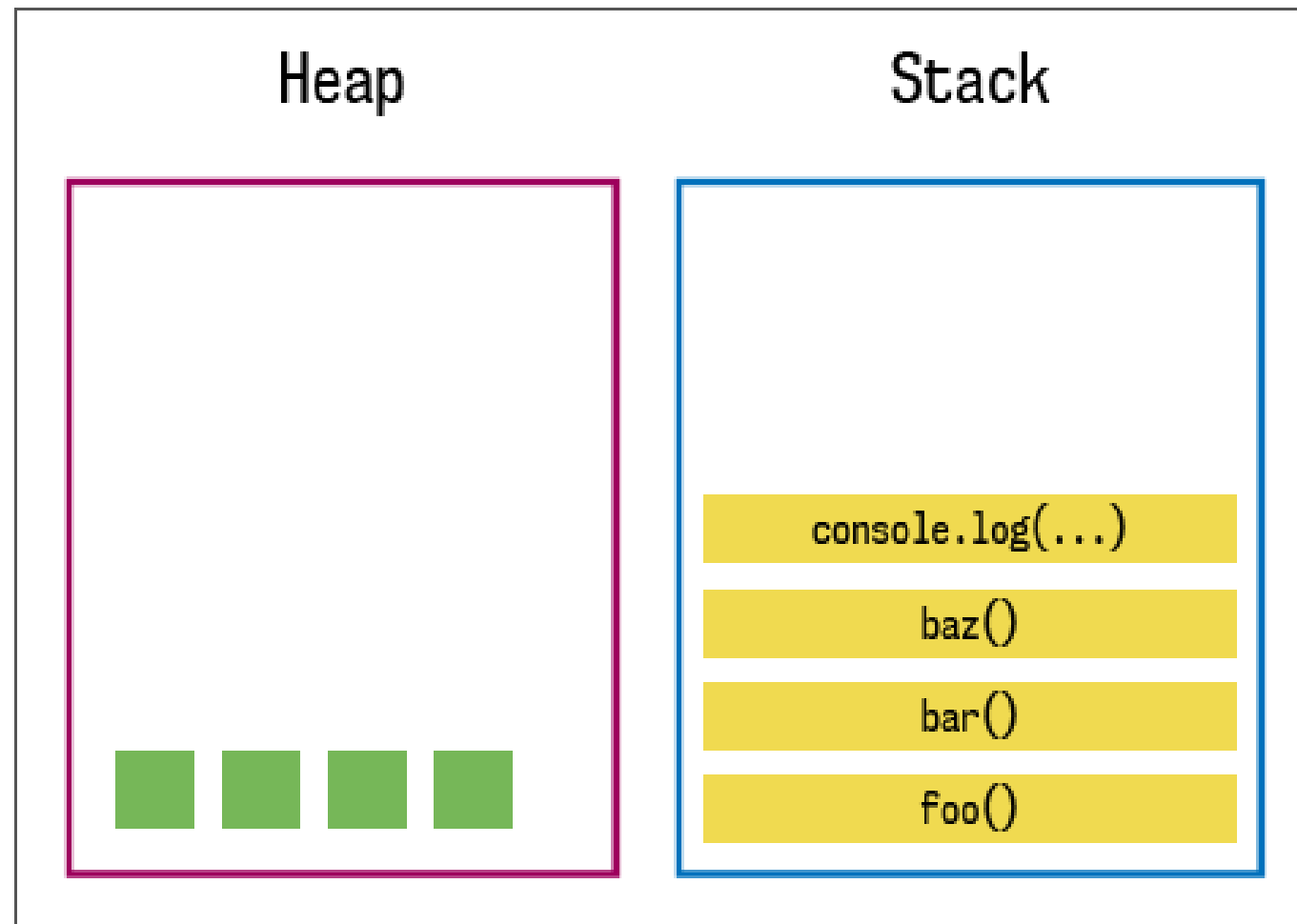
JAVASCRIPT

- Single threaded (Einer Thread)
- Never-blocking (Niemals blockierend)
- Asynchronous (Asynchron)
- Concurrent (Nebenläufig)

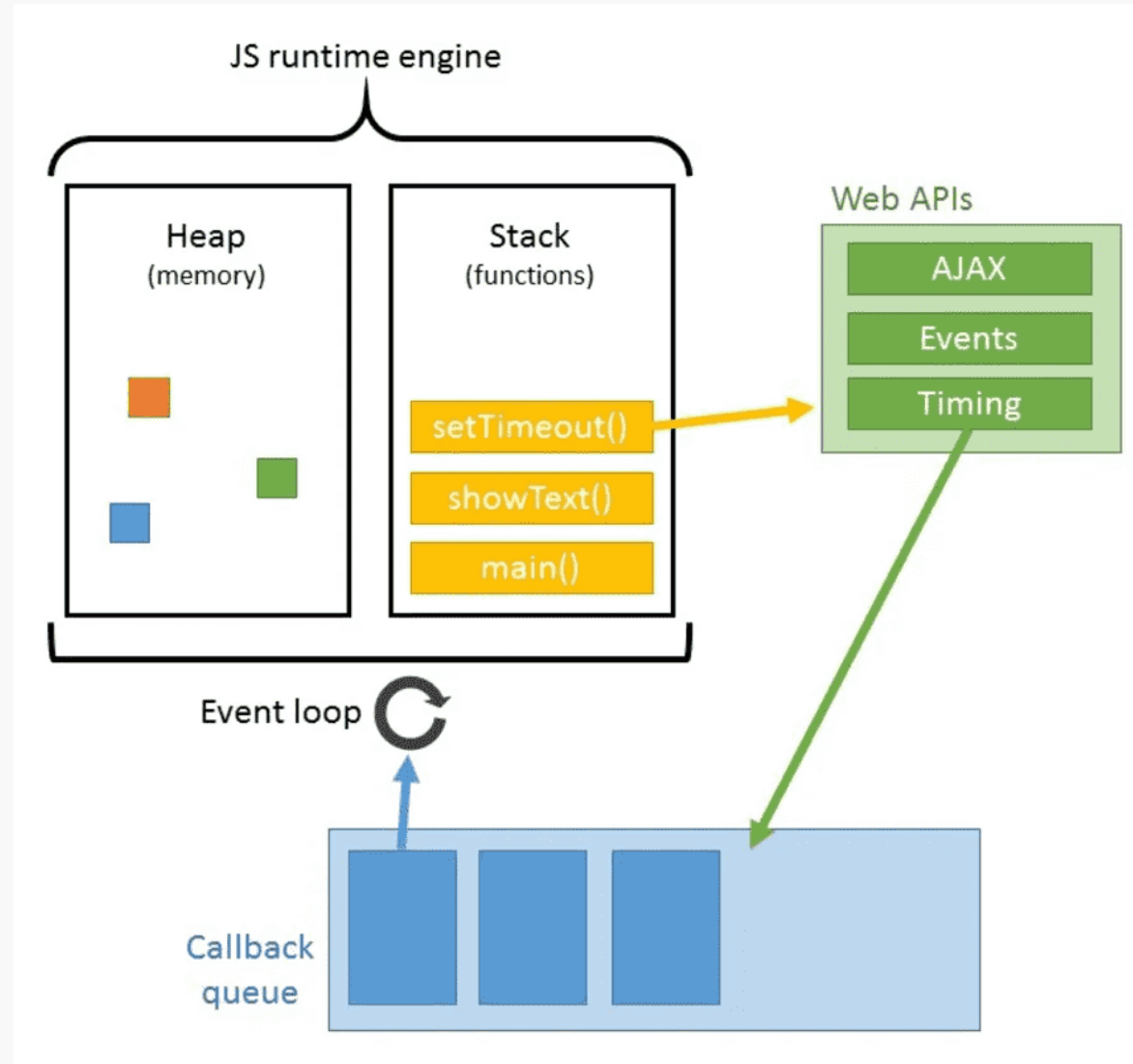
JS

- Call Stack
- Event Loop
- Callback Queue
- Other APIs

JavaScript Runtime



A WIDER LOOK



XML HTTP REQUESTS

```
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        // Typical action to be performed when the document is ready:
        document.getElementById("demo").innerHTML = xhttp.responseText;
    }
};
xhttp.open("GET", "filename", true);
xhttp.send();
```

XML HTTP REQUEST

Property	Description
onreadystatechange	Defines a function to be called when the readyState property changes
readyState	Holds the status of the XMLHttpRequest. 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
status	200: "OK" 403: "Forbidden" 404: "Page not found" For a complete list go to the Http Messages Reference
statusText	Returns the status-text (e.g. "OK" or "Not Found")

Full List : https://www.w3schools.com/tags/ref_httpmessages.asp

JQUERY AJAX

```
$(document).ready(function(){  
    $("button").click(function(){  
        $.ajax({url: "demo_test.txt", success: function(result){  
            $("#div1").html(result);  
        }});  
    });  
});
```


WITH AJAX GET & FAIL

```
$.get('demo_test.txt', data => {  
  console.log(data)  
}).fail(err => {  
  console.error(err)  
})|
```

FETCH API

```
fetch('https://yoursite.com')  
  .then(data => {  
    console.log(data)  
  })  
  .catch(err => {  
    console.error(err)  
  })
```