

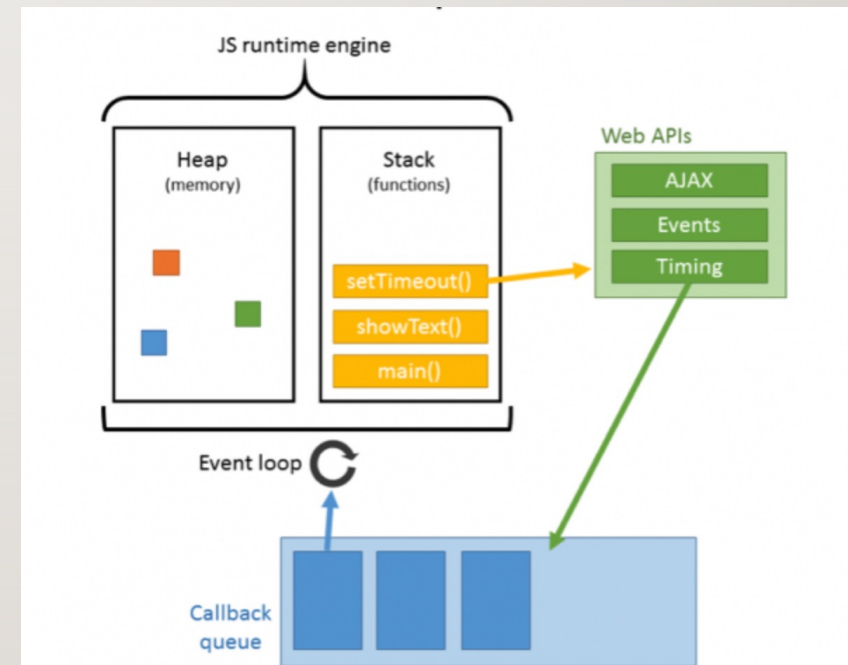
PROMISE

THEN, ASYNC, AWAIT, TRY, CATCH

АСИНХРОННОСТЬ В JAVA-SCRIPT

Java Script – это ...

- **Конкурентный** (concurrent) - в одно и то же время может происходить несколько явлений
- **Асинхронный** – эти явления могут происходить независимо и не дожидаться завершения друг друга
- **Однопоточный язык** – может обрабатывать только одну операцию за раз



ПАРАЛЛЕЛИЗМ В JS



- Достигается за счет среды выполнения, а не языка

CALLBACK HELL

Пример:

```
1  a(function (resultsFromA) {  
2      b(resultsFromA, function (resultsFromB) {  
3          c(resultsFromB, function (resultsFromC) {  
4              d(resultsFromC, function (resultsFromD) {  
5                  e(resultsFromD, function (resultsFromE) {  
6                      f(resultsFromE, function (resultsFromF) {  
7                          console.log(resultsFromF);  
8                      })  
9                  })  
10             })  
11         })  
12     })  
13 });
```

Недостатки:

- Нечитаемый код, растянут в ширину
- Результат выполнения доступен только на следующем уровне вложенности
- Обработку ошибок можно легко упустить
- Нельзя возвращать выражения с return

PROMISE

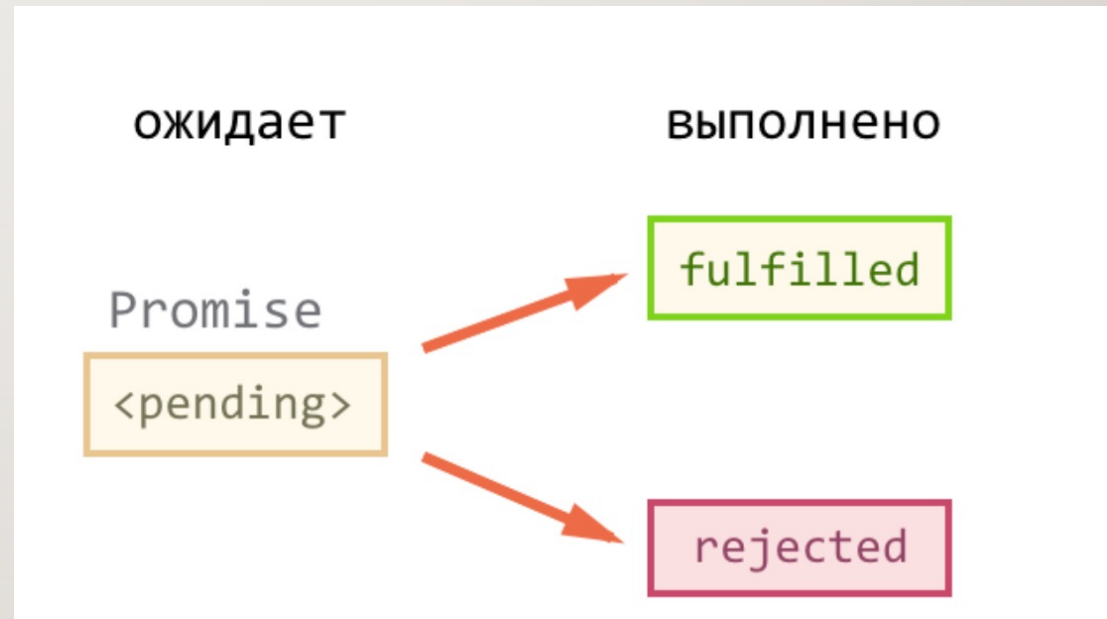
- это объект, который предоставляет доступ к значению, которое будет получено когда-то в будущем.

Promise позволяет установить значение только один раз.



СОСТОЯНИЯ PROMISE:

- Pending / ожидание
- Fulfilled / выполнено
- Rejected / отклонено



МЕТОДЫ PROMISE

I) PROMISE.ALL()

- Принимает итерируемый объект (например, массив) с несколькими промисами
- Возвращает единый промис, который выполнится в массив результатов всех промисов
- Если какой-нибудь из входных промисов отклонится, возвращаемый промис также сразу отклонится
- Использовать для выполнения связанных асинхронных операций, все из которых должны завершиться успешно

PROMISE.ALLSETTLED()

- Возвращает единый промис, который выполнится в массив результатов всех промисов
- Все входные промисы выполняются до конца
- Использовать для выполнения несвязанных асинхронных операций, либо когда важно отследить статус выполнения каждого промиса

ASYNC / AWAIT

- СИНТАКСИЧЕСКАЯ ОБЁРТКА ДЛЯ РАБОТЫ С PROMISE

- `Async` — ключевое слово, которое ставится перед функцией. После указания `async` функция всегда возвращает промис.
- `Await` тоже ключевое слово, которое заставит интерпретатор JavaScript ждать до тех пор, пока промис справа от `await` не выполнится. После чего оно вернёт его результат, и выполнение кода продолжится.

TRY / CATCH

- Конструкция try.. catch позволяет обрабатывать ошибки во время исполнения кода. Она позволяет запустить код и перехватить ошибки, которые могут в нём возникнуть.

