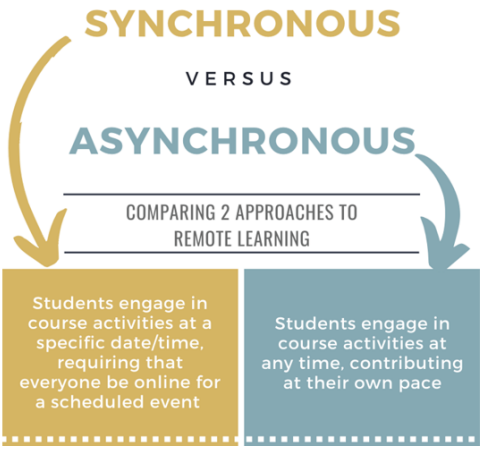
JavaScript Assíncrono

DIGITAL INNOVATION ONE

Definição

Assíncrono

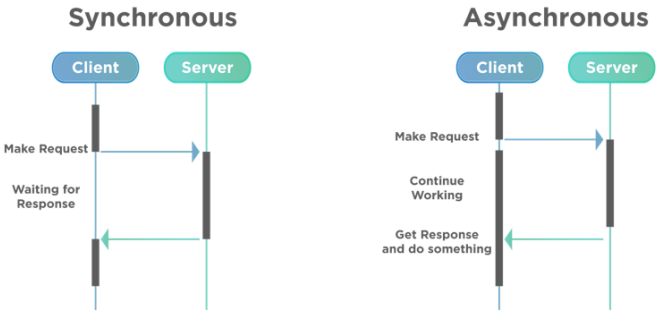
"Que não ocorre ou não se efetiva ao mesmo tempo."



DIGITAL INNOVATION ONE

Definição

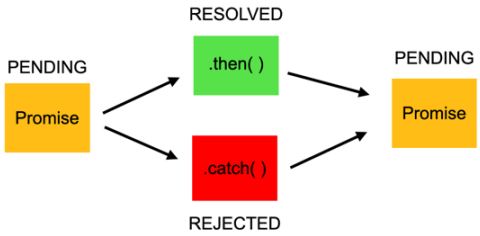
O Javascript roda de maneira **síncrona**.



DIGITAL INNOVATION ONE

Promises

Objeto de processamento assíncrono



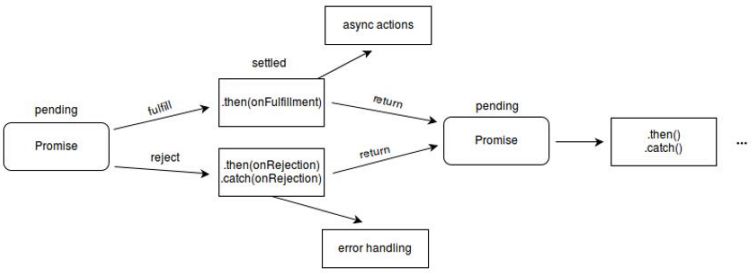
Inicialmente, seu valor é desconhecido. Ela pode, então, ser **resolvida** ou **rejeitada**.

DIGITAL INNOVATION ONE

Promises

Uma promise pode ter 3 estados

1) Pending **2)** Fulfilled **3)** Rejected



DIGITAL INNOVATION ONE

Promises

Estrutura

```
const myPromise = new Promise((resolve, reject) => {
    window.setTimeout(() => {
        resolve(console.log('Resolvida!'));
    }, 2000);
});
```

DIGITAL INNOVATION ONE

Promises

Manipulação

```
const myPromise = new Promise((resolve, reject) => {
    window.setTimeout(() => {
        resolve('Resolvida');
    }, 2000);
});

await myPromise
        .then((result) => result + ' passando pelo then')
        .then((result) => result + ' e agora acabou!')
        .catch((err) => console.log(err.message));

// Após 2 segundos, retornará o valor
// "Resolvida passando pelo then e agora acabou!"
```

DIGITAL INNOVATION ONE

Async/await

Funções assíncronas precisam dessas duas palavras chave.

```
async function resolvePromise() {
  const myPromise = new Promise((resolve, reject) => {
    window.setTimeout(() => {
        resolve('Resolvida');
    }, 3000);
});

  const resolved = await myPromise
        .then((result) => result + ' passando pelo then')
        .then((result) => result + ' e agora acabou!')
        .catch((err) => console.log(err.message));

  return resolved;
}
```

DIGITAL INNOVATION ONE

Async/await

Funções assíncronas também retornam Promises!

```
async function resolvePromise() {
  const myPromise = new Promise((resolve, reject) => {
    window.setTimeout(() => {
        resolve('Resolvida');
    }, 3000);
});

  const resolved = await myPromise
        .then((result) => result + ' passando pelo then')
        .then((result) => result + ' e agora acabou!')
        .catch((err) => console.log(err.message));

  return resolved;
}
```

```
> resolvePromise()
< ▶ Promise {<pending>}
> await resolvePromise()
< "Resolvida passando pelo then e agora acabou!"
```

DIGITAL INNOVATION ONE

Async/await

Utilizando try...catch

```
async function resolvePromise() {
    const myPromise = new Promise((resolve, reject) => {
        window.setTimeout(() => {
            resolve('Resolvida');
        }, 3000);
    });

  let result;

    try {
    result = await myPromise
        .then((result) => result + ' passando pelo then')
        .then((result) => result + ' e agora acabou!')
  } catch(err) {
    result = err.message;
  }

    return result;
}
```