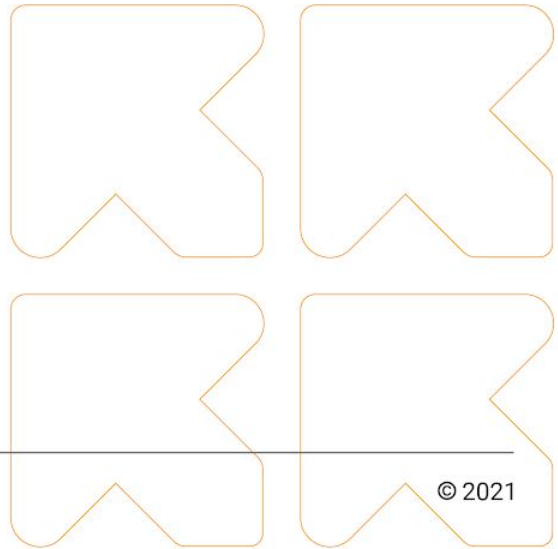
fazztrack

Week 2

Javascript Introduction 3

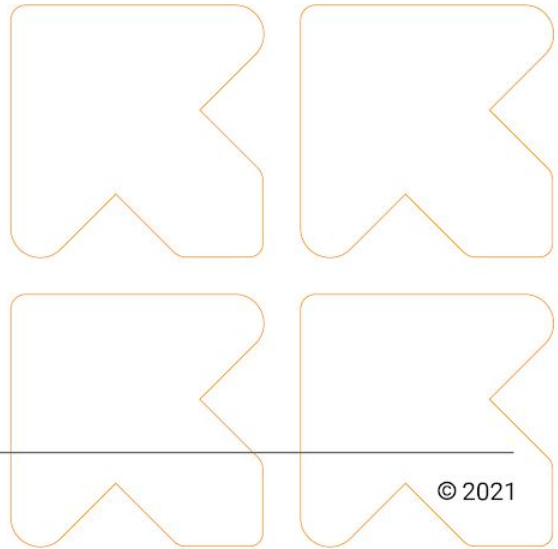
Rules

- Absence
- Follow the rules
- Ask us anything (bootcamp matters in private)
- Speak for yourself first
- Trainer availability
- Independent
- Hard work
- Do your best
- Continuous self improvement



Objective

- Asynchronous
- Promise
- Callback Asynchronous
- Async Await
- Try Catch



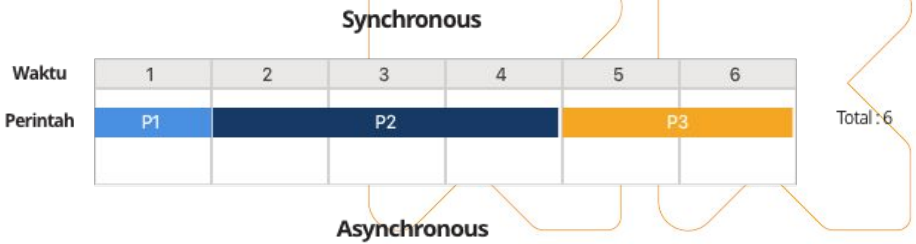
fazztrack

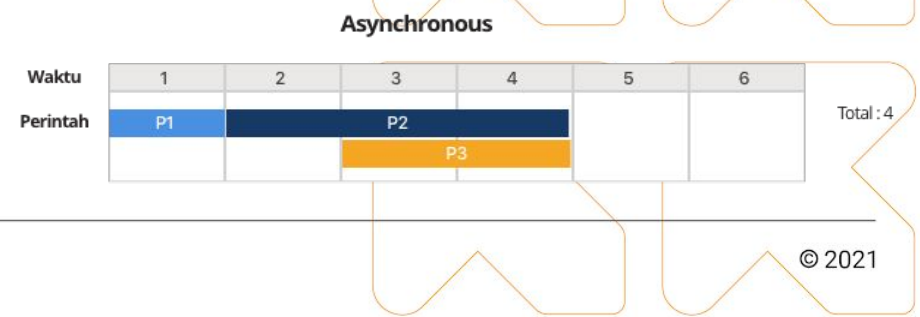
Asynchronous

Asynchronous

- Asynchronous VS Synchronous
  - Synchronous = perintah dieksekusi satu persatu sesuai urutan kode yang anda tuliskan
  - Asynchronous = hasil eksekusi atau output tidak selalu berdasarkan urutan kode tetapi berdasarkan waktu proses

Perintah	Waktu
P1	1
P2	3
P3	2



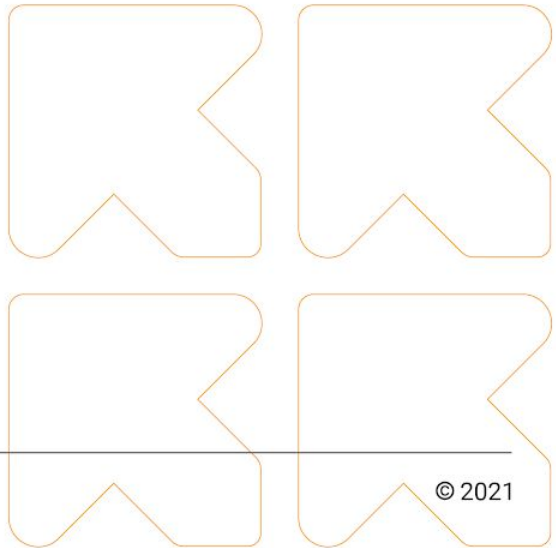


Asynchronous in Javascript

- Javascript secara default mengeksekusi perintah secara synchronous
- Tetapi ada beberapa case yang dimana code javascript akan tereksekusi dengan asynchronous seperti : event, timer, request ajax, listener, interaksi user, dll

Handle Asynchronous

- Callback Function/Asynchronous
- Promise
- Async/Await



fazztrack

Callback Asynchronous

Callback Asynchronous

- Callback function atau callback (biasa disingkat dengan cb) adalah salah satu metode yang paling umum yang digunakan untuk menghandle return value dari operasi asynchronous.
- Callback sendiri adalah sebuah regular function (yang biasanya anonymous) dan ditaruh di argumen paling belakang dari sebuah asynchronous function. Layaknya function biasa, callback juga dapat menerima parameter dan mengembalikan value.

https://medium.com/coderupa/panduan-komplit-asynchronous-programming-pada-javascript-part-2-callback-3a717df6cfdf

Kekurangan Callback Asynchronous

- Code sulit dibaca dan sulit untuk di maintenence
- Tidak ada error handling

```
a((resultFromA) => {
  b(resultFromA, (resultFromB) => {
    c(resultFromB, (resultFromC) => {
      d(resultFromC, (resultFromD) => {
        e(resultFromD, (resultFromE) => {
          f(resultFromF, (resultFromF) => {
            console.log(resultFromF)
          })
        })
      })
    })
  })
})
```

fazztrack

Promise

Promise

- Promise merupakan perwakilan dari sebuah nilai yang belum tentu diketahui nilainya saat promise dibuat. Promise memungkinkan pengguna untuk menghubungkan fungsi handler dengan keberhasilan atau kegagalan aksi asynchronous.

```
let janjian = new Promise((resolve, reject) => {
  let success = false
  if(success){
    resolve('berhasil')
  }else{
    reject(new Error('janji dibatalkan'))
  }
})
```

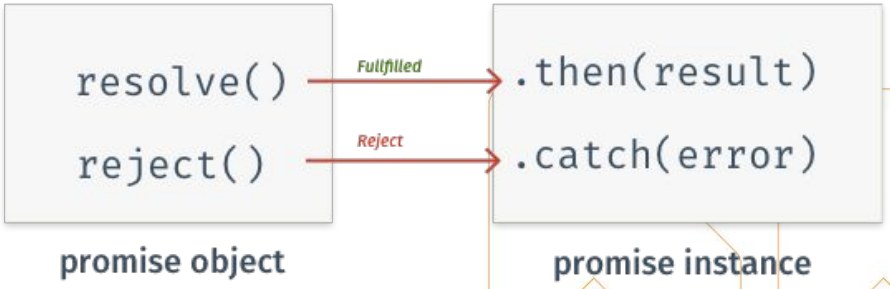
Promise

- Promise ada di salah satu state ini:
  - **Pending** → keadaan awal, tidak terpenuhi atau ditolak
  - **Fulfilled** → artinya operasi selesai dengan sukses.
  - **Rejected** → artinya operasi gagal.

Promise

- Ketika menggunakan promise bisa menggunakan then dan catch

```
janjian.then((result) => {
  console.log(result)
}).catch((error) => {
  console.log(error)
})
```



https://medium.com/coderupa/panduan-komplit-asynchronous-programming-pada-javascript-part-3-promise-819ce5d8b3c

fazztrack

Async/Await

Async/Await

- Async/await adalah fitur yang hadir sejak ES2017. Fitur ini mempermudah kita dalam menangani proses asynchronous.

```
async function helloWorld(){
  let result = await doAsync()
  console.log(result)
}
```

Keterangan :

async → mengubah function menjadi synchronous

await → menunda eksekusi hingga proses asynchronous selesai,

https://medium.com/coderupa/panduan-komplit-asynchronous-programming-pada-javascript-part-4-async-await-fc504c344238

Try Catch dan Finally

- Untuk mengatasi Error (error handling) pada async/await bisa menggunakan try catch
  - Try
    biasanya kita sisipkan code javascript yang mungkin terjadi error
  - Catch
    blok inilah yang akan menangkap error yang terjadi pada blok Try apabila pada blok Try si error muncul.
  - Finally
    blok ini digunakan untuk apapun yang terjadi pada blok Try, baik itu error atau tidak, akan selalu dijalankan.

```
async function helloWorld(){
  try{
    let result = await doAsync()
    console.log(result)
  }catch(error){
    console.log(error)
  }finally{
    console.log('Proses selesai')
  }
}
```

fazztrack

Live Coding

Task Javascript Introduction

1. Buatlah sambungan program menggunakan :
   a. then catch
   b. try catch
   untuk mengecek hari kerja dari kode Asynchronous disamping dan jelaskan penggunaan then catch dan try catch dengan memberikan komentar di bawahnya

```
const cekHariKerja = (day) => {
  return new Promise((resolve, reject) => {
    setTimeout(() => {
      const dataDay = ['senin', 'selasa', 'rabu', 'kamis', 'jumat']
      let cek = dataDay.find((item) => {
        return item === day
      })
      if(cek){
        resolve(cek)
      }else {
        reject(new Error('Hari ini bukan hari kerja'))
      }
    }, 3000)
  })
}
```

Task Javascript Introduction

2. Buat program menggunakan callback function untuk melanjutkan dan menampilkan semua bulan menggunakan method map

   Contoh :
   getMonth(showMonth?)

```
const getMonth = (callback) => {
  setTimeout(() => {
    let error = false
    let month = ['Januari', 'Februari', 'Maret', 'April',
                 'Mei', 'Juni', 'Juli', 'Agustus', 'September',
                 'Oktober', 'November', 'Desember']
    if (!error) {
      callback(null, month)
    } else {
      callback(new Error('Sorry Data Not Found'), [])
    }
  }, 4000)
}
```

Task Javascript Introduction

3. Buatlah 2 program bebas dengan menggunakan promise seperti soal nomor 1
4. Buatlah program menggunakan method fetch untuk menampilkan semua name (hanya name nya saja) dari REST API dibawah ini
   https://jsonplaceholder.typicode.com/users
   Gunakan debugger console bawaan browser Chrome untuk melihat hasilnya