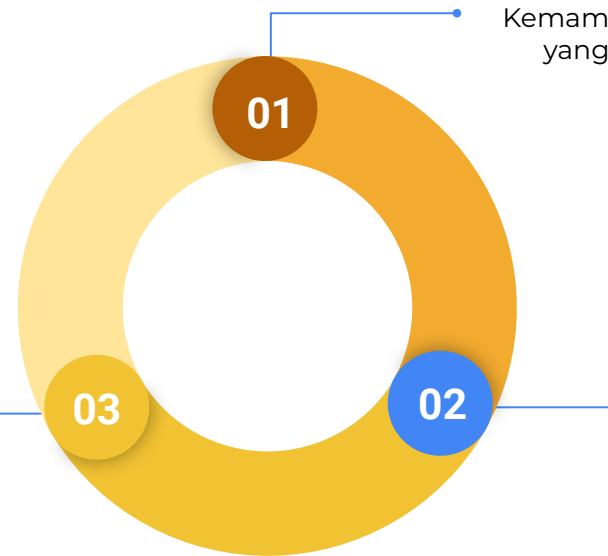


# Frontend Javascript Challenge

**Silver** Chapter 3

### ACCEPTANCE CRITERIA

Kriteria pengumpulan challenge yang harus kamu penuhi untuk dapat melewati chapter



### SKILL METRICS

Kemampuan teknis yang akan kamu pelajari

### DELIVERY

Hal yang akan kamu lakukan untuk dapat melewati chapter

### SKILL METRICS

(Kemampuan teknis yang akan kamu pelajari)

1. NODEJS
2. Object-Oriented Programming

### DELIVERY

(Hal yang akan kamu lakukan untuk dapat melewati chapter)

1. Memahami konsep Web Server dan Node.JS sebagai runtime environment
2. Mampu melakukan instalasi Node.js dan package manager dengan menggunakan NPM atau Yarn
3. Mampu membuat project sederhana menggunakan NodeJS
4. Memahami konsep OOP
5. Mampu membuat implementasi OOP

### CRITERIA

(Kriteria pengumpulan challenge yang harus kamu penuhi untuk dapat melewati chapter)

1. Menggunakan Node.JS sebagai runtime environmnet
2. Membuat implementasi konsep OOP

Pada challenge chapter kali ini, kamu akan diminta **membuat sebuah project menggunakan Node.js dan menyelesaikan studi kasus tentang OOP didalam project tersebut.**

Studi kasus dibawah telah dilengkapi dengan instruksi, penjelasan problem, kriteria function, serta contoh pengeksekusian/penggunaan function tersebut beserta expected output yang akan dihasilkan pada function yang kamu buat nantinya.



1

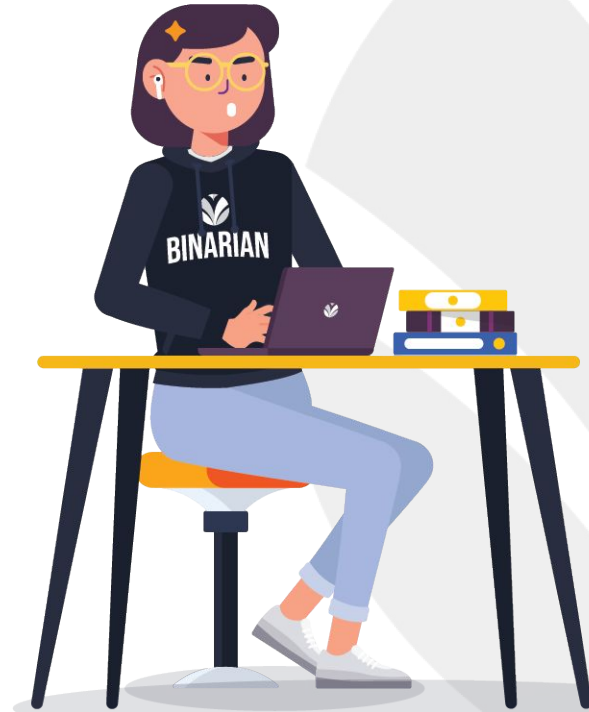
Buatlah sebuah function dengan nama **Ship** yang berfungsi untuk menghitung apakah isi atau berat kapal itu lebih banyak muatan-nya atau lebih banyak crew-nya.

Function ini akan menerima 2 parameter, yaitu :

1. **Draft** => Perkiraan berat kapal berdasarkan seberapa rendahnya di dalam air
2. **Crew** => Jumlah awak kapal yang ada didalam kapal.

Dan terdapat sebuah deklarasi variabel **EmptyShip** yang nantinya akan diisi oleh nilai **Draft** dan **Crew** (input).

**Catatan:** Awak kapal dianggap menambah 1,5 unit draft, dan jika kapal yang memiliki draft lebih dari 20 akan dianggap lebih banyak muatannya ketimbang awak kapalnya.



## Sample Case

```

var emptyShip = new Ship(draft, crew);

function Ship(draft, crew) {
  this.draft = draft;
  this.crew = crew;
  // tuliskan code kalian disini
}

// EXPECTED RESULT
// Ketika output function itu menghasilkan angka diatas 20
// console.log("Banyak nih muatan kapalnya")

// Ketika output function itu menghasilkan angka dibawah 20
// console.log("Lebih banyak muatan orangnya..")
```



2

Buatlah sebuah class **Animal** yang berisi constructor dengan nama variabel *name*, *age*, *legs*, *species*, & *status*, dst. (lihat slide 11).

Tugas kalian adalah mendefinisi tiga class berikut ini yang merupakan Inheritance dari class **Animal**.

### 1. Shark

Function constructor untuk **Shark** harus menerima total tiga argumen dalam urutan sebagai berikut (*name*, *age*, *status*).

**Catatan:** Variabel *legs* di **shark** harus terisi jumlah kaki **0** dan *species*-nya “**shark**”.

### 2. Cat

Function constructor untuk **Cat** harus menerima total tiga argumen dalam urutan sebagai berikut (*name*, *age*, *status*).

**Catatan:** Variabel *legs* di **cat** harus terisi jumlah kaki **4** dan *species*-nya “**cat**”.



2

### 3. Dog

Function constructor untuk **Dog** harus menerima total empat argumen dalam urutan sebagai berikut (***name, age, status, master***). **Master** adalah nama dari pemilik **Dog** tersebut yang tentunya berisi *string*.

**Catatan:** Variabel *legs* di **dog** harus terisi jumlah kaki **4** dan *species*-nya **"dog"**. Method *introduce* untuk *Dog* mirip dengan yang lainnya, yang membedakan adalah *Dog* memiliki method sendiri yang bernama **GreetMaster** yang dimana tidak menerima argumen dan me-return **"Hello (*insert\_master\_name\_here*)"**, ganti yang didalam kurung dengan nama master yang tentunya *string*.



## Sample Case

```
class Animal {
  constructor(name, age, legs, species, status) {
    this.name = name;
    this.age = age;
    this.legs = legs;
    this.species = species;
    this.status = status;
  }
  introduce() {
    return `Hello, my name is ${this.name} and I am ${this.age} years old.`;
  }
}

class Shark extends Animal {
  constructor(/* masukkan parameter */) {
    super(/* panggil parent class menggunakan argument yang benar */);
  }
}

class Cat extends Animal {
  // tuliskan code kalian disini
}

class Dog extends Animal {
  // tuliskan code kalian disini
}

// Contoh output (Shark)
// nama Hiu, umur 10, kaki 0, spesies Shark, status Berenang.
```

Nah apabila kamu telah selesai mengerjakan semua soal, maka persiapkan diri kamu untuk melakukan presentasi dan menjawab pertanyaan yang akan diberikan oleh fasilitator ya.

Facilitator akan memberikan kamu pertanyaan acak yang bertujuan untuk mengetes pemahaman koding kamu.

**Selamat mencoba dan Have Fun Exploring!**



# Terima Kasih