S1 Z4

Arrays and Objects

Iffe (question ☺)

```
let dataObject = {
    id:1,
    data: 'example data'
}

var proxy = (function(foo){
    return {
        getData: function(){
            return foo;
        },
        setData: function(val) {
            foo.data = val;
        }
    }
})(dataObject.data);

console.log(proxy.getData()); //{id: 1, data: 'example data'}
proxy.setData('changed data');
console.log(proxy.getData()); //{id: 1, data: 'changed data'}

console.log(dataObject); //changed data
```

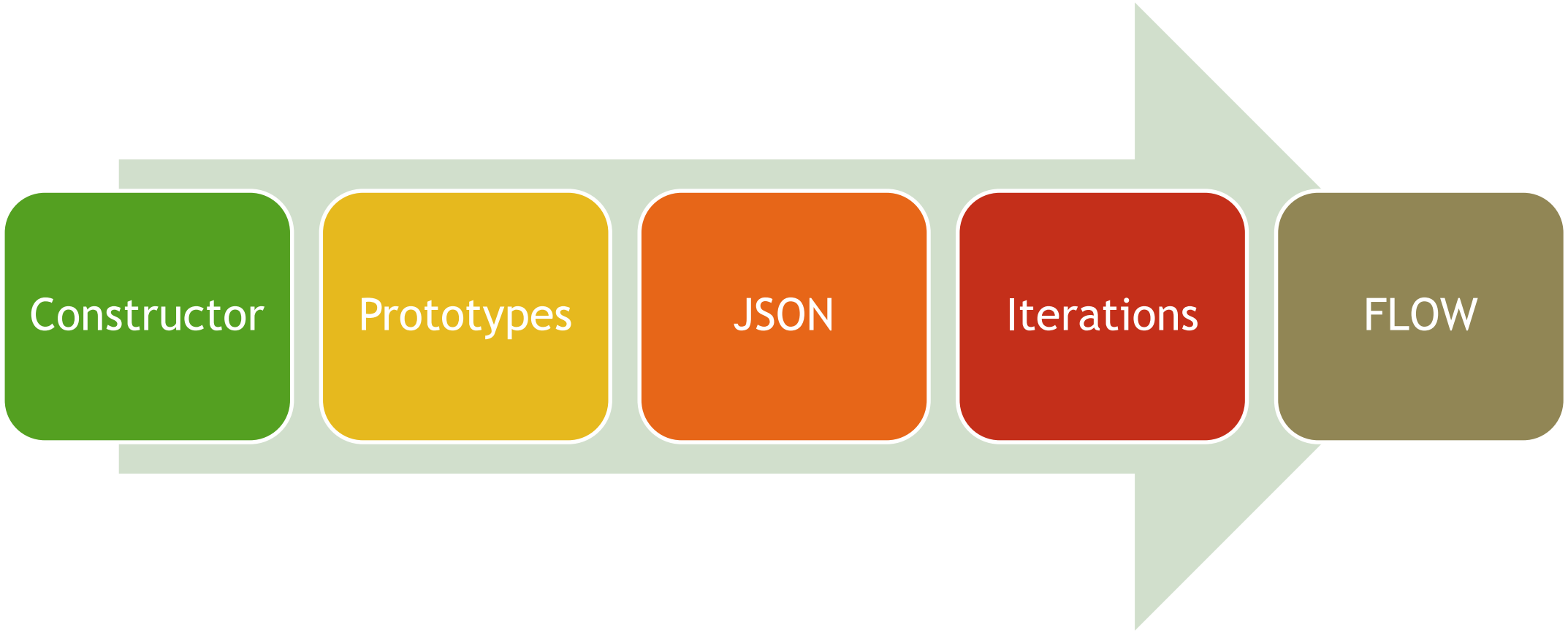
Iffe (question ☺)

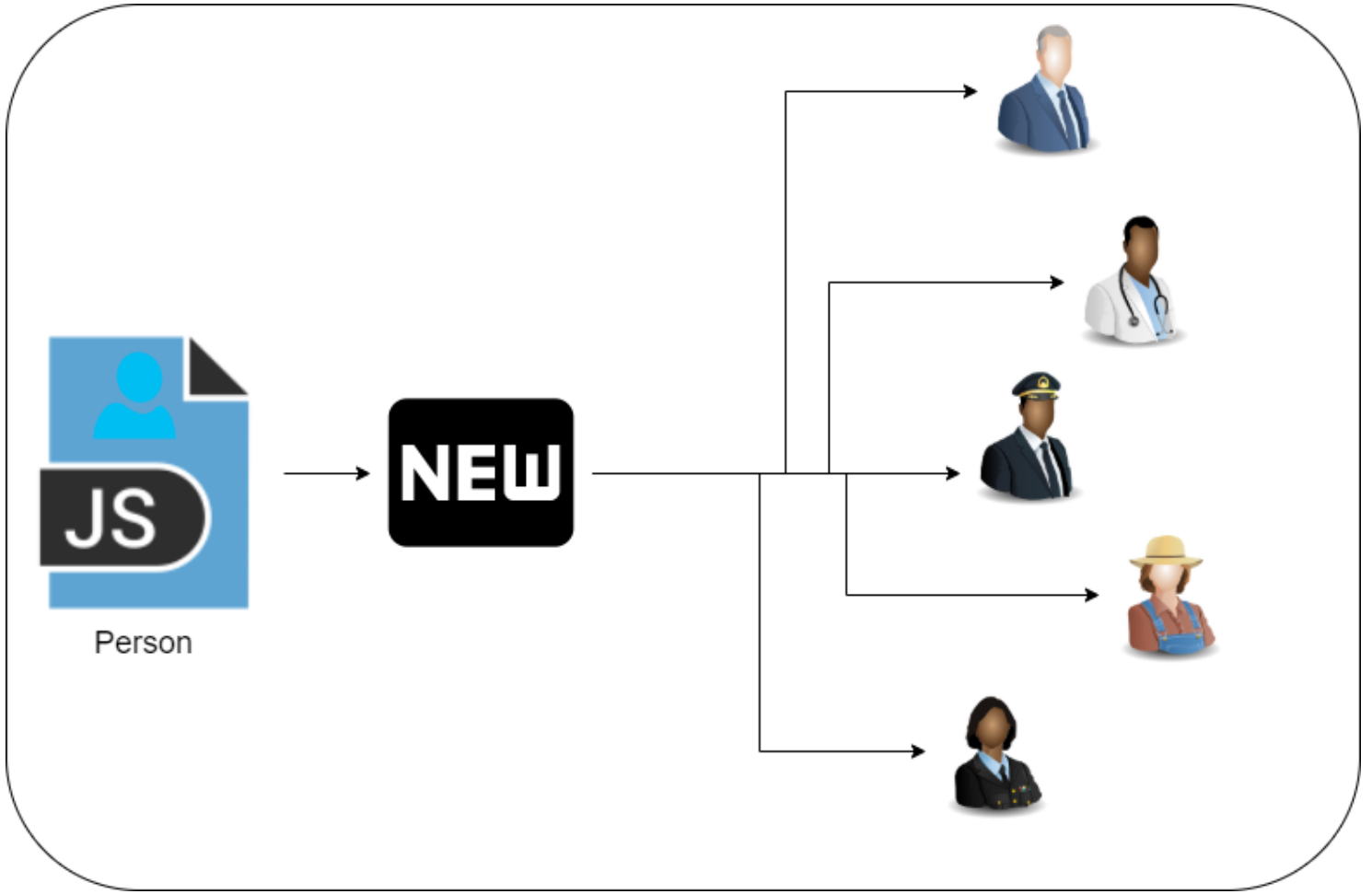
```
let dataObject = {
    id:1,
    data: 'example data'
}

var proxy = (function(foo){
    return {
        getData: function(){
            return foo;
        },
        setData: function(val) {
            foo.data = val;
        }
    }
})(Object.assign({},dataObject));

console.log(proxy.getData()); //{id: 1, data: 'example data'}
proxy.setData('changed data’); //{id: 1, data: 'changed data'}
console.log(proxy.getData());

console.log(dataObject.data); //example data
```

assign

```
let obj1 = { x: 1, y: 2, z: 3, f: () => {} };
let obj2 = { z: 4 };
let result;
result = Object.assign({}, obj1, obj2);
console.log(result); //{x: 1, y: 2, z: 4, f: f}
result = Object.assign({}, obj2, obj1);
console.log(result); //{z: 3, x: 1, y: 2, f: f}
result = Object.assign({ x: 0 }, obj2, obj1);
console.log(result); //{x: 1, z: 3, y: 2, f: f}
result = Object.assign(obj1, obj2);
console.log(result); //{x: 1, y: 2, z: 4, f: f}
```

Arrays and Objects




JS
Person
NEW

Constructor

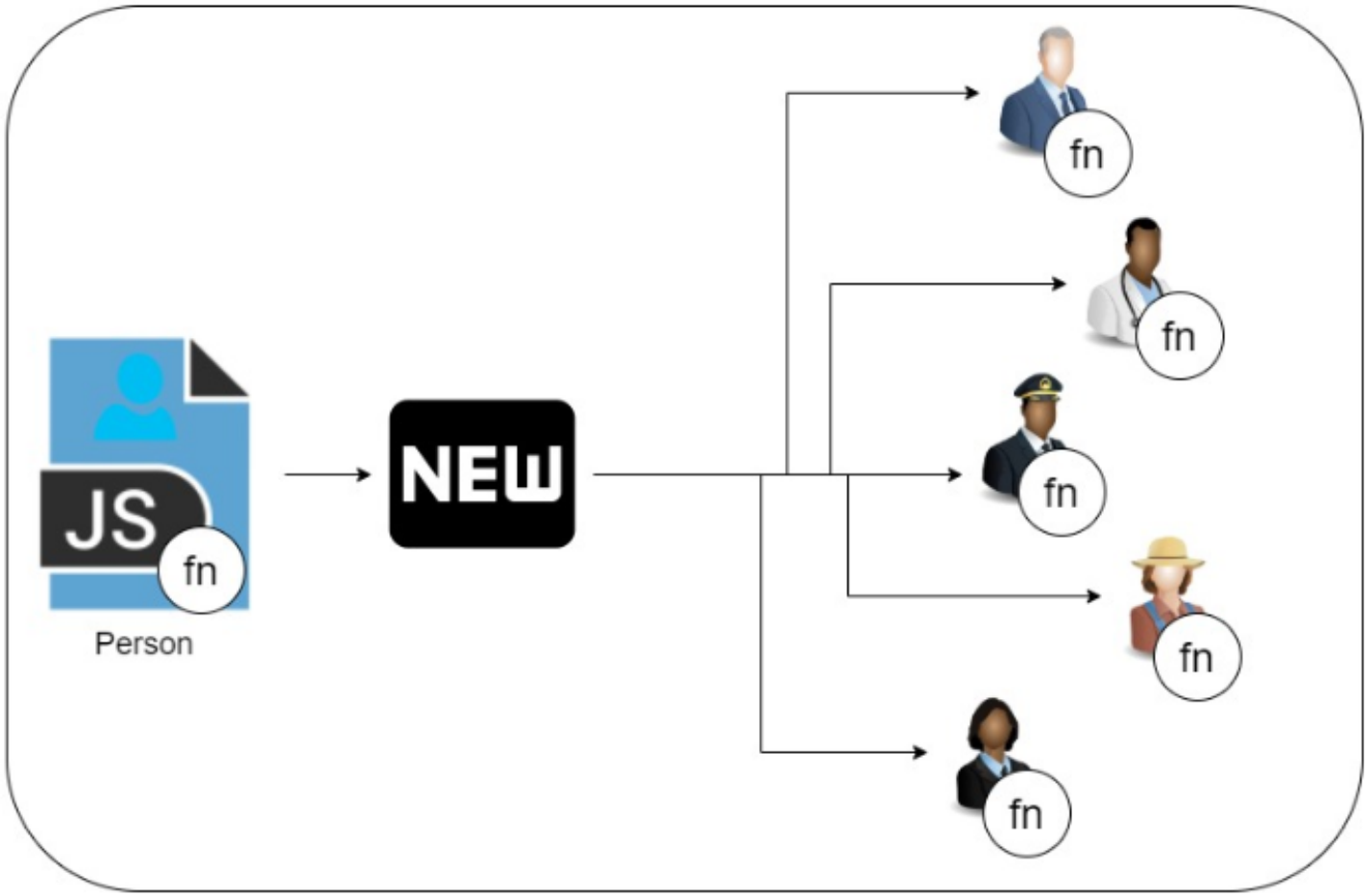
```
function Person() {}

let karol = new Person();

console.log(karol); //Person
console.log(typeof karol); //object
```

Constructor

```
function Person() {
  console.log(this);
  return 1;
}

let karol = new Person();
console.log(karol); //Person

let adam = Person();
console.log(adam); //1

let tomek = Person;
console.log(tomek); //ƒ Person()

let pawel = new tomek();
console.log(pawel); //Person
```

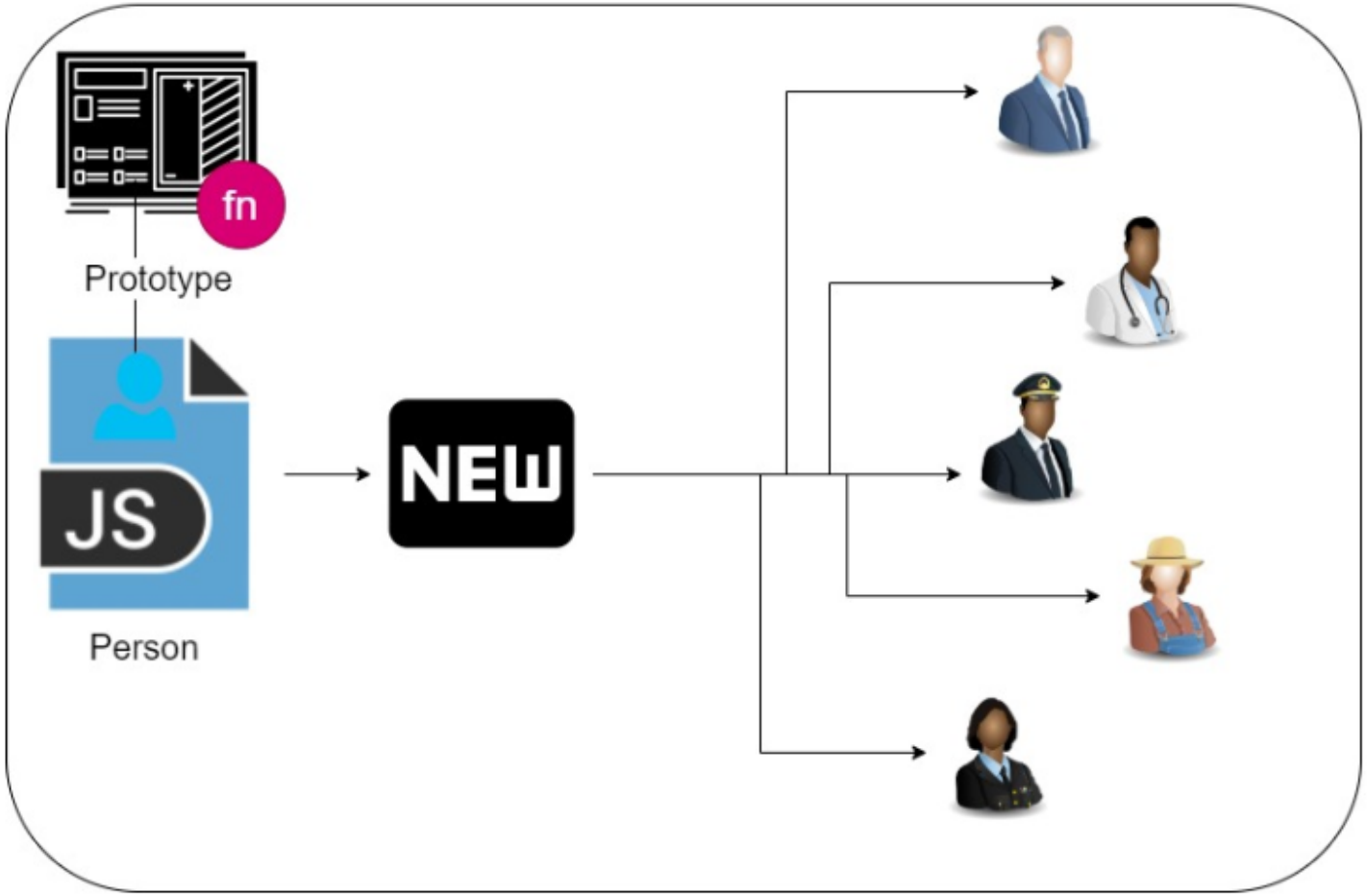
Constructors

```
function Person(firstName, lastName) {
  this.firstName = firstName;
  this.lastName = lastName;
}
let karol = new Person("Karol", "Rogowski");
console.log(karol);//Person {firstName: 'Karol', lastName: 'Rogowski'}
let adam = new Person("Adam", "Goral");
console.log(adam); //Person {firstName: 'Adam', lastName: 'Goral'}
```


JS
fn
Person
NEW
fn
fn
fn
fn
fn

Constructor

```
function Person(firstName, lastName) {
  this.firstName = firstName;
  this.lastName = lastName;
  this.sayHello = () =>
    console.log("Hello from " + this.firstName + " " + this.lastName);
}

let karol = new Person("Karol", "Rogowski");
let adam = new Person("Adam", "Goral");
console.log(karol); //Person {firstName: 'Karol', lastName: 'Rogowski', sayHello: ƒ}
karol.sayHello(); //Hello from Karol Rogowski
```


fn
Prototype
JS
Person
NEW

Prototype

```
function Person(firstName, lastName) {
  this.firstName = firstName;
  this.lastName = lastName;
}

Person.prototype.sayHello = function () {
  console.log("Hello from " + this.firstName + " " + this.lastName);
};

let karol = new Person("Karol", "Rogowski");
let adam = new Person("Adam", "Tur");

console.log(karol, adam); //Person {firstName: 'Karol', lastName: 'Rogowski'} Person {firstName:
'Adam', lastName: 'Tur'}

karol.sayHello(); //Hello from Karol Rogowski
```

Prototype

```
function Person(firstName, lastName){
    this.firstName = firstName;
    this.lastName = lastName;
}

let karol = new Person('Karol','Rogowski');

Person.prototype.sayHello = function() {
    console.log( 'Hello from ' + this.firstName + ' ' + this.lastName);
}

console.log(karol); //Person {firstName: 'Karol', lastName: 'Rogowski'}
karol.sayHello(); //Hello from Karol Rogowski
```

Prototype

```
function Person(firstName, lastName){
    this.firstName = firstName;
    this.lastName = lastName;
}

let karol = new Person('Karol','Rogowski');

console.log(karol);
karol.sayHello(); //TypeError: karol.sayHello is not a function

Person.prototype.sayHello = function() {
    console.log( 'Hello from ' + this.firstName + ' ' + this.lastName);
}
```

Prototype

```
function Person(firstName, lastName){
    this.firstName = firstName;
    this.lastName = lastName;
}

let karol = new Person('Karol','Rogowski');

Person.prototype.sayHello = () => {
    console.log( 'Hello from ' + this.firstName + ' ' + this.lastName);
    console.log(this); //{}
}

console.log(karol);
karol.sayHello(); //Hello from undefined undefined
```

Prototype

```
String.prototype.showMe = function(){
    console.log('Hello world from '+this);
}

'Karol Rogowski'.showMe(); //Hello world from Karol Rogowski
```

Prototype

```
Number.prototype.getValueDescription = function(){
    return "My value is: " + this
}

console.log((4).getValueDescription()); //My value is: 4
```

Prototype

```
function Demo(){
    console.log('Demo function result');
}

Function.prototype.customRun = function(){
    console.log('Custom run begin');
    this();
    console.log('Custom run end');
}

Demo.customRun();
//Custom run begin
//Demo function result
//Custom run end
```

Prototype

```
function Demo() {
  for (let i = 0; i < 1000; i++) {
    for (let j = 0; j < 1000; j++) {
      for (let k = 0; k < 1000; k++) {}
    }
  }
}

Function.prototype.measureExecTime = function () {
  let ts = new Date().getTime();
  this();
  let te = new Date().getTime();
  return te - ts;
};

console.log("Execution took " + Demo.measureExecTime() + " milisecends");
//Execution took 699 milisecends
```

JSON

```
let person = {
    id: 1,
    name:'Karol Rogowski'
}

console.log(person);
//{id: 1, name: 'Karol Rogowski'}

console.log(JSON.stringify(person));
//{"id":1,"name":"Karol Rogowski"}
```

JSON

```
let people = [{
        id: 1,
        name:'Karol Rogowski'
    },{
        id: 2,
        name:'Jan Kowalski'
    },{
        id: 3,
        name:'Robert Lewandowski’
    }]

console.log(people);
console.log(JSON.stringify(people));
```

JSON

```
let personJSON = `{
    "id":1,
    "name":"Karol Rogowski"
}`;

let person = JSON.parse(personJSON);
console.log(person);
```

JSON

```
let peopleJSON = `[
        {
            "id":1,
            "name":"Karol Rogowski"
        },
        {
            "id":2,
            "name":"Jan Kowalski"
        },
        {
            "id":3,
            "name":"Robert Lewandowski"
        }
    ]`

let people = JSON.parse(peopleJSON);
console.log(people);
```

Array Iteration

```
let people = [
    {
        innerId: 'dfr458hj',
        name: 'Karol Rogowski',
        birthYear: 1985,
        sayHello: function(){console.log(this.name+ ' says hello')}
    },
    {
        innerId: 'plo745as',
        name: 'Jan Kowalski',
        birthYear:1980,
        sayHello: function(){console.log(this.name+ ' says hello')}
    },
    {
        innerId: 'qaz390pl',
        name: 'Robert Lewandowski',
        birthYear: 1988,
        sayHello: function(){console.log(this.name+ ' says hello')}
    }
]
```

Array Iteration

```
people.forEach(p => console.log(p));

people.forEach((p,i)=>console.log(i+':'+ p.name));

people.forEach(p => p.sayHello());
```

Array Iteration

```
console.log(people.filter(p=> p.birthYear > 1980));

console.log(people.every(p=> p.birthYear > 1980));

console.log(people.every(p=> p.birthYear >= 1980));
```

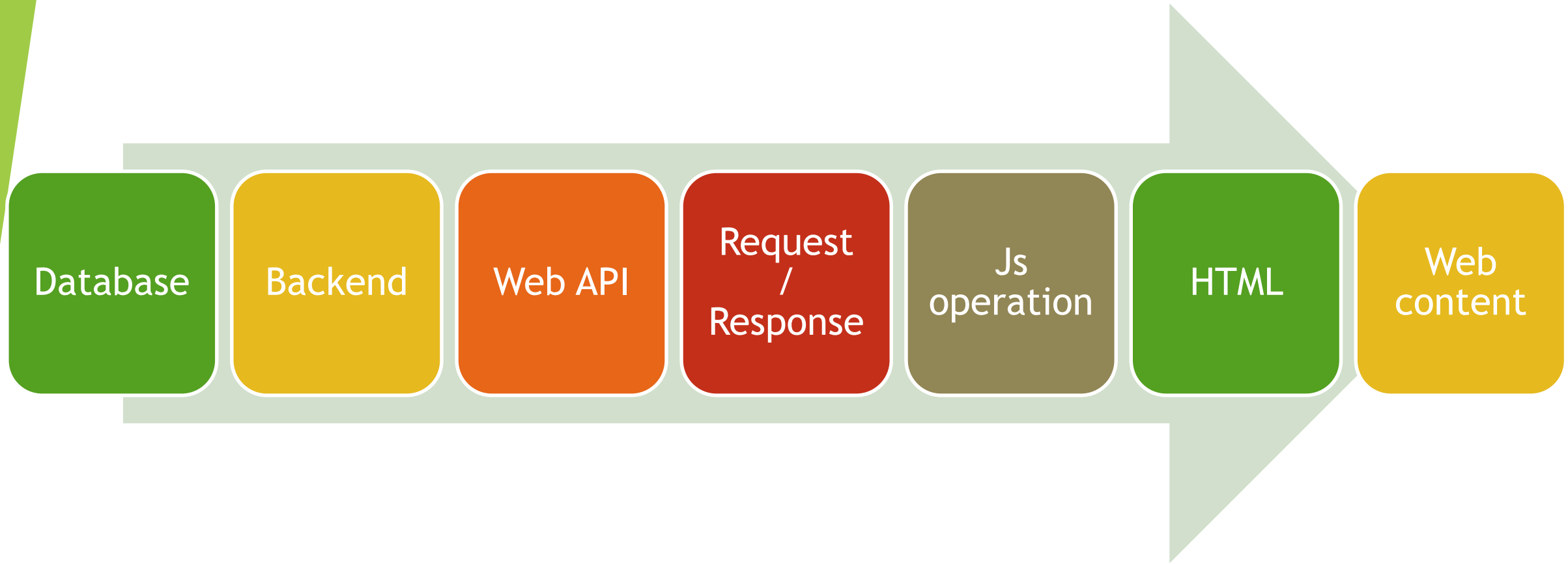
Array Iteration

```
console.log(people.map((p,i)=>i + ':' + p.name));

console.log(people.find(p=>p.birthYear != 1985));
```

Array Iteration

```
function Person(firstName, lastName, id){
    this.id = id;
    this.firstName = firstName;
    this.lastName = lastName;
}

console.log(people.map((p,i)=> new Person(p.name.split(' ')[0],
                                          p.name.split(' ')[1],
                                          i)));
```

Flow – big picture



Flow “API”

```
let apiObject = {
    getPeople: ()=>`[
        {
            "id":1231,
            "name":"Karol Rogowski"
        },
        {
            "id":2123,
            "name":"Jan Kowalski"
        },
        {
            "id":3111,
            "name":"Robert Lewandowski"
        }
    ]`
}
```

FLOW “mappings”

```
function Person(firstName, lastName, id){
    this.id = id;
    this.firstName = firstName;
    this.lastName = lastName;
}

let result = JSON.parse(apiObject.getPeople());

console.log(result.map((p)=> new Person(p.name.split('')[0],
                                        p.name.split(' ')[1],p.id)));
```