HAZZA INSTITUTE OF TECHNOLOGY

iqracity

MERN – ES6 + React

Module: 12 ES6 - II

Outline

Module: 12 ES6 - II

HAZZA INSTITUTE OF TECHNOLOGY

- Understand concepts of methods Array Function map()
- Understand concepts of methods Array Function reduce()
- Understand concepts of methods Array Function: filter()
- Understand concepts of methods Array Function: find() and findIndex()
- Understand concepts Classes in ES6
- Understand concepts of Inheritance in ES6
- Understand concepts Working with Promises

HAZZA INSTITUTE OF TECHNOLOGY

Array Function map()

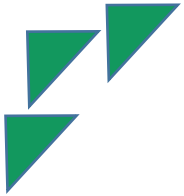
Array Function: map()

HAZZA INSTITUTE OF TECHNOLOGY

It iterates the array for us and we can pass a callback function to perform some operation on the each array item. The updated values can be returned by the callback function to create a new array.

Syntax:

```
arr.map((item) => {
   //Callback function body.
})
```



HAZZA INSTITUTE OF TECHNOLOGY

Array Function reduce()

Array Function: reduce()

HAZZA INSTITUTE OF TECHNOLOGY

Just like map(), reduce() also iterates through the entire array and it accepts a callback function to perform some action on the array element. The difference here is that reduce() passes the result of the callback from one iteration to the next one. This callback result is called accumulator. The accumulator can be pretty much anything (integer, string, object or even an array) and must be instantiated and passed when calling reduce().

Syntax:

```
arr.reduce((acc, item) => {
   //callback function body
}, acc_default_value)
```

HAZZA INSTITUTE OF TECHNOLOGY

Array Function: filter()

Array Function: filter()

HAZZA INSTITUTE OF TECHNOLOGY

It iterates through the array to create a new array. You can decide which elements should be added in the new array based on some conditions.

Syntax:

```
arr.filter(item => {
   //Return true/false to add/skip the current item
})
```

HAZZA INSTITUTE OF TECHNOLOGY

Array Function: find() and findIndex()

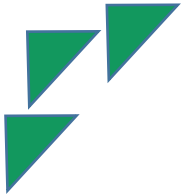
HAZZA INSTITUTE OF TECHNOLOGY

Array Function: find() and findIndex()

ES6 introduced 2 new methods to search for elements inside the array.

- find(): It is used to search for an element in the array that matches some condition. It returns the first element that matchesthe condition.

- findIndex(): It is quite similar to the find() method. The difference is that findIndex() method returns the index of the element instead of the element itself.



HAZZA INSTITUTE OF TECHNOLOGY

Classes in ES6

Classes in ES6

HAZZA INSTITUTE OF TECHNOLOGY

To create object factories, you can use the class keyword. Just like you would do in other programming languages like JAVA etc.

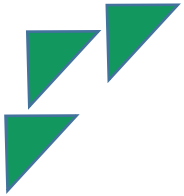
Classes in ES6

HAZZA INSTITUTE OF TECHNOLOGY

Syntax:

```
class ClassName {
   constructor() {
      // Initialize the properties here
   }
   //Methods outside constructor
   methods1 = () => {
   //Method body
}
}
```

This is just syntax-sugar. Behind the scenes everything still works the same.



HAZZA INSTITUTE OF TECHNOLOGY

Inheritance in ES6

Inheritance in ES6

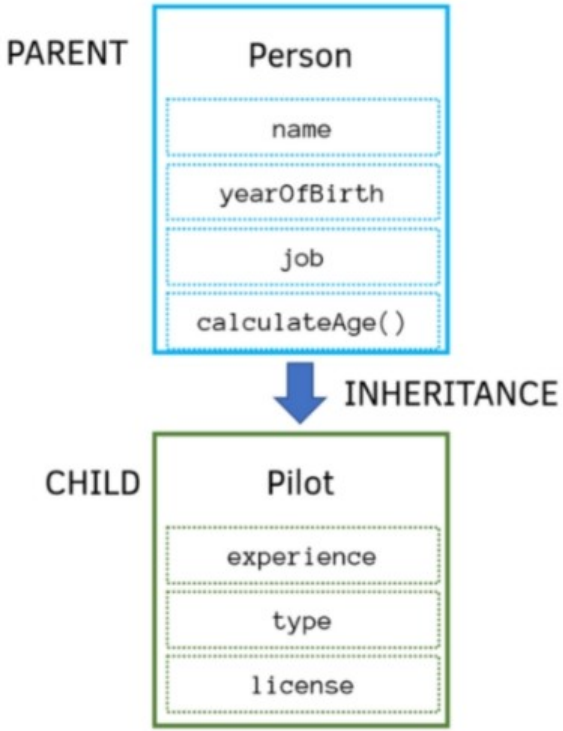
HAZZA INSTITUTE OF TECHNOLOGY

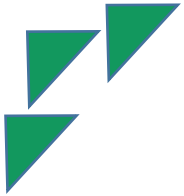
ES6 provides us with a keyword called extends to inherit classes.

Syntax:

```
class ChildClass {
   // Class body
}
class ChildClass extends ParentClass {
   // Class body
}
```

HAZZA INSTITUTE OF TECHNOLOGY





HAZZA INSTITUTE OF TECHNOLOGY

Working with Promises

HAZZA INSTITUTE OF TECHNOLOGY

How to create Promises?

Syntax:

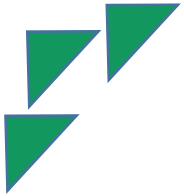
```
const mPromise = new Promise((resolve, reject) => {
   // Promise body
   //Call resolve() when the operation is complete.
   //Call reject() when the operation is failed.
})
```

HAZZA INSTITUTE OF TECHNOLOGY

"then()" and "catch()" Methods

then() method is called when the resolve() is executed. It receives data passed in the resolve() method as arguments.

catch() method is called when the reject() method is executed. It receives the data passed in the reject() method as arguments.



HAZZA INSTITUTE OF TECHNOLOGY

Callbacks and Promises

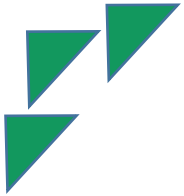
HAZZA INSTITUTE OF TECHNOLOGY

Callbacks and Async Operations

In JavaScript the code is executed line-by-line in a sequence so when we run a parallel operation or asynchronous operation like fetching data from backend, JavaScript doesn't waits for the response it simply executes the next line of code. We give the asynchronous operation a function to call when it is completed. This function is called a Callback Function.

HAZZA INSTITUTE OF TECHNOLOGY

Introduction to Promises



Introduction to Promises

A promise is used to handle the asynchronous result of an operation. It defers the execution of a code block until an asynchronous request is completed. This way, other operations can keep running without interruption.

A promise has 3 states:

- Pending: It means the operation is going on.
- Fulfilled: It means the operation was completed.
- Rejected: It means the operation did not complete and an error can be thrown.

Summary

Module: 12 ES6 - II

HAZZA INSTITUTE OF TECHNOLOGY

- Understand concepts of methods Array Function map()
- Understand concepts of methods Array Function reduce()
- Understand concepts of methods Array Function: filter()
- Understand concepts of methods Array Function: find() and findIndex()
- Understand concepts Classes in ES6
- Understand concepts of Inheritance in ES6
- Understand concepts Working with Promises