GlintsAcademy
Powered by BINAR ACADEMY

Review
ES6, OOP, Async,
JS Library/Framework

Isumi
Batam, Jan 2020

BINAR ACADEMY

BINAR ACADEMY

Material:

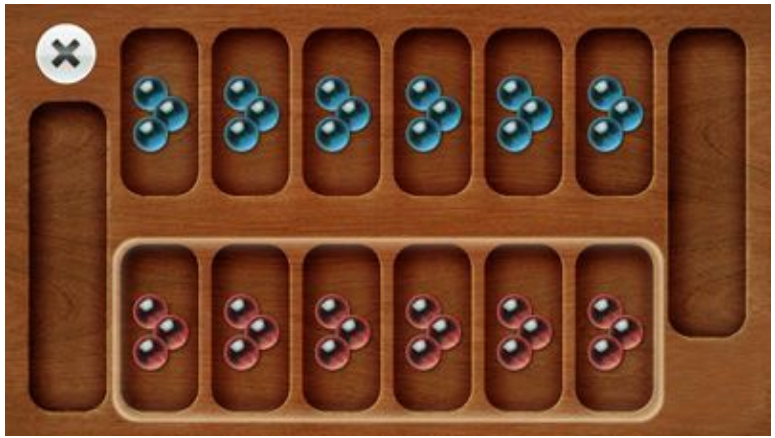
- ES6
- OOP
- Async
- ReactJS

QA..?

:: NOTES ::

Remember!

- Global vs Local variable (scope)
- Explore: Functional Programming
- Closure?



BINAR ACADEMY

Closure?

When the function has finished the execution, the scope is usually destroyed.

```
function buildName(name) {
    let greeting = "Hello, " + name;
    return greeting;
}
```

Remember:

- The function scope is created for a function call, not for the function itself
- Every function call creates a new scope

So, how about closure?

Example: Using Closure

```
function buildName(name) {
   let greeting = "Hello, " + name + "!";
   let sayName = function() {
       console.log(greeting);
       let welcome = greeting + " Welcome!";
       console.log(welcome);
   };
   return sayName;
}

let sayMyName = buildName("Isumi");
sayMyName();
```

Remember:

- Closure are nested function which has access to the outer scope
- After the outer function is returned, by keeping a reference to the inner function (the closures) we prevent the outer scope to be destroyed.

So, closure is a function which has access to the variable from another function's scope.

For more info, read this article

GlintsAcademy
Powered by BINAR ACADEMY

CODE REVIEW

BINAR ACADEMY

BINAR ACADEMY