

<b>Started on</b>	Friday, 1 September 2023, 8:27 PM
<b>State</b>	Finished
<b>Completed on</b>	Friday, 1 September 2023, 8:35 PM
<b>Time taken</b>	7 mins 16 secs
<b>Marks</b>	23.00/23.00
<b>Grade</b>	<b>10.00</b> out of 10.00 (100%)

**QUESTION 1**

Correct

Mark 3.00 out of 3.00

What functions are correct?

```
//A
const generatePowerA = (a, b) => { return a ** b; };
//B
let generatePowerB = (a, b) => { return a ** b; };
//C
const generatePowerC = function(a, b){ return a ** b; };
//D
let generatePowerD = function(a, b){ return a ** b; };
//E
const generatePowerE = a, b => { return a ** b; };
//F
let generatePowerF = (a, b) => a ** b;
//G
const generatePowerG = (a, b) => { a ** b; };
//H
let generatePowerH = (a, b) => {
  a ** b;
};
```

Select one or more:

- ☐ 1. H
- ☐ 2. G
- ☒ 3. F ✓
- ☒ 4. C ✓
- ☒ 5. D ✓
- ☒ 6. B ✓
- ☐ 7. E
- ☒ 8. A ✓

**QUESTION 2**

Correct

Mark 1.00 out of 1.00

Using an array method without parameters, delete the last element of a given array:

```
const users = ["Andriy", "Iryna", "Ihor", "Petro"];
```

Please, do not create any additional variables in your answer.

Code of your answer:

**QUESTION 3**

Correct

Mark 1.00 out of 1.00

Using the methods of arrays, add two numbers 1000 and 1350 to the beginning of the given array?

```
const salary = [550, 900, 1800, 2200];
```

Please, do not create any additional variables.

Code of your answer:

**QUESTION 4**

Correct

Mark 1.00 out of 1.00

Using the array methods, create a new array that contains the elements of the array *arr1* and the elements of the array *arr2*.

```
const arr1 = ["User1", "User2", "User3"];  
const arr2 = [1, 2, 3];  
// your code ---> ["User1", "User2", "User3", 1, 2, 3]
```

Please, do not create any additional variables in your answer.

Code of your answer:



**QUESTION 5**

Correct

Mark 1.00 out of 1.00

For which of the given data types is the correct definition: "value was not assigned"?

Select one:

- ☐ a. string
- ☐ b. null
- ☒ c. undefined ✓
- ☐ d. object
- ☐ e. boolean

**QUESTION 6**

Correct

Mark 1.00 out of 1.00

What is the result of executing this code?

```
'b' + 'a' + +'a' + 'a';
```

Select one:

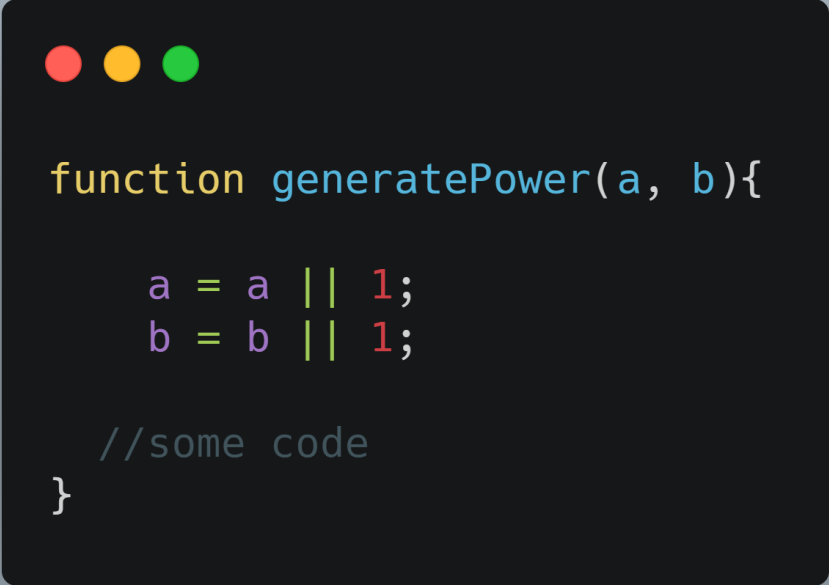
- ☐ a. null
- ☐ b. "ba1a"
- ☐ c. "baaa"
- ☐ d. undefined
- ☐ e. NaN
- ☒ f. "baNaNa" ✓

## QUESTION 7

Correct

Mark 1.00 out of 1.00

What syntax are we emulating with this code?



```
function generatePower(a, b){  
  
    a = a || 1;  
    b = b || 1;  
  
    //some code  
}
```

Select one:

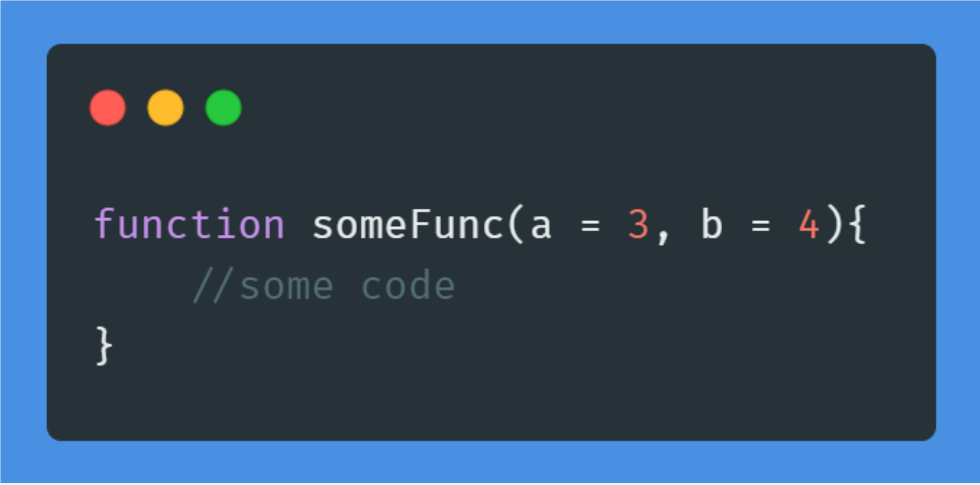
- ☒ a. default parameters ✓
- ☐ b. IIFE
- ☐ c. logical comparison
- ☐ d. arrow function
- ☐ e. smart condition

**QUESTION 8**

Correct

Mark 1.00 out of 1.00

What syntax were we emulating with this code?



```
function someFunc(a = 3, b = 4){  
  //some code  
}
```

Select one:

- ☐ a. `a = 3;`  
`b = 4;`
- ☐ b. `a = b = 3;`
- ☐ c. `a = a && 3;`  
`b = b && 4;`
- ☒ d. `a = a || 3;` ✓  
`b = b || 4;`
- ☐ e. `[a, b] = [3, 4];`

## QUESTION 9

Correct

Mark 1.00 out of 1.00

Why this code does not work?

```
function findAstro(humans){  
  const humLen = humans.length;  
  for(let i = 0; i < humLen; i++){  
    if(humans[i].position === 'astrounaut'){  
      break;  
    }  
  }  
  
  console.log(`Astrounaut is in ${i} index`);  
  
  return i;  
}
```

Select one:

- ☐ a. We cannot use a loop to step through the elements of an array.
- ☐ b. *Break* is only for switch statement
- ☐ c. This code is correct
- ☒ d. Variable *i* in block scope and we can't see it after loop ✓

## QUESTION 10

Correct

Mark 1.00 out of 1.00

What is the name of a special array that contains all the arguments passed to the function when it is called?

Select one:

- ☐ a. spread
- ☐ b. parameters
- ☐ c. array
- ☒ d. arguments ✓

**QUESTION 11**

Correct

Mark 1.00 out of 1.00

Which statements are valid for a given function?

```
function square(a) {  
  result = a ** 2;;  
  console.log("Your square: " + result);  
}
```

Select one or more:

- ☐ a. Function Expression
- ☒ b. named function ✓
- ☐ c. returns the value
- ☐ d. has no parameters
- ☒ e. Function Declaration ✓
- ☒ f. the function can be called before its declaration ✓

**QUESTION 12**

Correct

Mark 1.00 out of 1.00

Which statements are valid for a given function?

```
let add = function(a, b) {  
  return a + b;  
};
```

Select one or more:

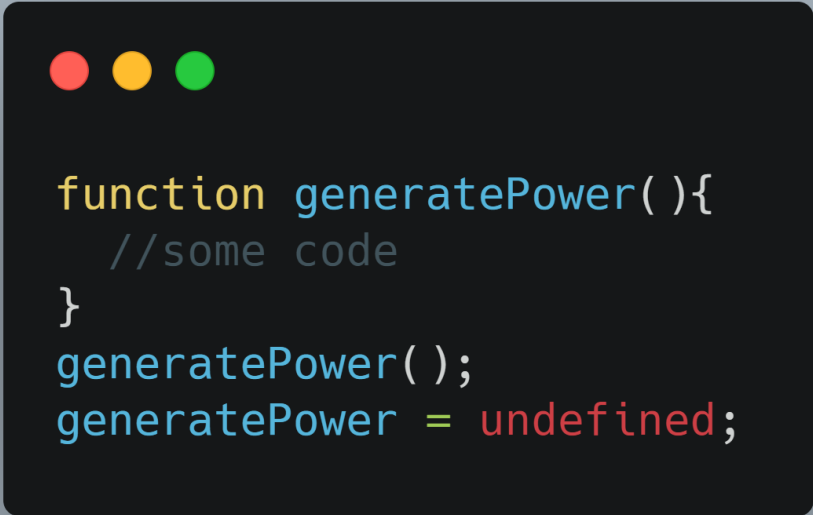
- ☐ a. the function can be called before its declaration
- ☐ b. has no parameters
- ☒ c. returns the value ✓
- ☐ d. Function Declaration
- ☒ e. Function Expression ✓

**QUESTION 13**

Correct

Mark 1.00 out of 1.00

What syntax are we emulating with this code?



```
function generatePower(){  
    //some code  
}  
generatePower();  
generatePower = undefined;
```

Select one:

- ☐ a. logical comparison
- ☒ b. IIFE ✓
- ☐ c. smart condition
- ☐ d. arrow function
- ☐ e. default parameters



**QUESTION 14**

Correct

Mark 2.00 out of 2.00

What expression can be converted to false?

Select one or more:

- ☒ a. 0 ✓
- ☐ b. "false"
- ☒ c. undefined ✓
- ☒ d. null ✓
- ☐ e. true
- ☒ f. NaN ✓
- ☒ g. "" or " " or `` (empty strings) ✓

**QUESTION 15**

Correct

Mark 1.00 out of 1.00

Template strings are enclosed by

Select one:

- ☒ a. backtick `` ✓
- ☐ b. double quotes "
- ☐ c. curly brackets {}
- ☐ d. single quotes '

**QUESTION 16**

Correct

Mark 1.00 out of 1.00

Arrow function can be used as constructors

Select one:

- ☐ True
- ☒ False ✓

**QUESTION 17**

Correct

Mark 1.00 out of 1.00

An arrow function has not its own:

- super
- arguments
- this

Select one:

- ☒ True ✓
- ☐ False

**QUESTION 18**

Correct

Mark 1.00 out of 1.00

Self-Executing Anonymous Function is not the same as Immediately Invoked Function Expression

Select one:

- ☐ True
- ☒ False ✓

**QUESTION 19**

Correct

Mark 1.00 out of 1.00

Spread syntax "expands" an array into its elements, while rest syntax collects multiple elements and "condenses" them into a single element.

Select one:

- ☒ True ✓
- ☐ False

**QUESTION 20**

Correct

Mark 1.00 out of 1.00

Which of the following is true about Symbol?

Select one or more:

- ☒ a. there is a global registry of symbols ✓
- ☐ b. iterated by a for ... in loop
- ☒ c. is primitive type ✓
- ☒ d. `Symbol("user") !== Symbol("user")` ✓
- ☒ e. allow to create "hidden" properties of objects ✓

◀ Tasks \* JS for React

Jump to... ▾

Slides. Object oriented design ▶