1. 電子郵件

justme11012@gmail.com

1. 嗨嗨，你的名子是?

林潔君

1. What is OS? (Operation System)?

Operating system (OS) is a program that manages a computer’s resources, especially the allocation of those resources among other programs. Typical resources include the central processing unit (CPU), computer memory, file storage, input/output (I/O) devices, and network connections. Management tasks include scheduling resource use to avoid conflicts and interference between programs. Unlike most programs, which complete a task and terminate, an operating system runs indefinitely and terminates only when the computer is turned off.

1. What is Linux?

Linux is an operating system, just like windows and Mac OS, it manages the communication between the software and the hardware in your device. But what makes Linux so special is because it’s open source software. The code used to create Linux is free and available to the public to view and edit. There are many distributions and customizable resource of Linux can use in word processors, web browsers, and many different applications. Linux users also can choose core components, such as which system displays graphics, and other user-interface components. Companies and individuals choose Linux for their servers because it's secure, flexible, and you can receive excellent support from a large community of users.

1. What is variable in JavaScript?  
   What's the different between var, let and const?

What is variable scope?

Variables are symbolic names for values in our applications. It’s a way of storing and keeping

track of information in a program so that it can be later used and manipulated. There are three ways to create or declare a variable, the const, let, var keyword. But they have different rules about how you use them and change their values in your program to prevent a variable might accidentally get changed or overwritten with a different value further down in your program.

- var: The oldest keyword to declare a variable in JavaScript. It has the Global scoped or function scoped which means variables defined outside the function can be accessed globally, and variables defined inside a particular function can be accessed within the function.

- let: The let keyword is an improved version of the var keyword. It is introduced in the ES6 or EcmaScript 2015. These variables has the block scope. It can’t be accessible outside the particular code block ({block}).

- const: The const keyword has all the properties that are the same as the let keyword, except the user cannot update it and have to assign it with a value at the time of declaration. These variables also have the block scope. It is mainly used to create constant variables whose values can’t be changed once they are initialized with a value.

In summary:  
|Keyword |Scope |Hoisting |Reassign |

|var |Function Scope |Yes |Yes |  
|let |Block Scope |No |Yes |  
|const |Block Scope |No |No |

Scope of variables refers to the accessibility of a particular variable within the program. There are 4 types of scopes, global scope, local scope, block scope, and function scope.

The variables in global scope are declared outside a block (the curly braces{} like “function” or “if” statement) called global variables and they can be accessed anywhere in the program. The variables in local scope are declared inside a block called local variables and they can be accessed only in the block.

After ES6 (ECMAScript 6) in 2015, JavaScript provides “let” and “const” keyword for strict situation and also delivery the block scope and function scope to distinguish different cases. Function scope is the variables declared in function declaration and user can only access the variables in the function, no matter how many conditional statements included. Block scope is the variables declared inside the block (the curly braces{} , it’s not necessary to declare with “if” or “function” keywords) and user can only access the variables in the block.

In the real situation, the code will be complicate, take 3 examples as following, and you can understand well.

**let is block scope**

function example() {

  if (true) {

**let** y = 20;

    console.log(y); // Output: 20

  }

  console.log(y);

// ReferenceError: y is not defined

}

example();

**var is function scope**function example() {

  if (true) {

**var** x = 10;

    console.log(x); // Output: 10

  }

  console.log(x); // Output: 10

}

example();

**const is block scope and cannot be reassigned**

function example() {

  const z = 30;

  console.log(z); // Output: 30

  z = 40;

// TypeError: Assignment to constant variable

}

example();

<https://www.linkedin.com/pulse/var-vs-let-const-easiest-explanation-ever-asit-rohan-dass>

<https://www.simplilearn.com/tutorials/javascript-tutorial/scope-of-variables-in-javascript>

1. What is value in JavaScript?  
   What's the different between primitive data type and non-primitive data type in JavaScript?

Value as a property or a part of data of a variable is the real thing we need to use in JavaScript and that retrieves from elements or passed as an argument to a function. Values in JavaScript can be of various types, including primitive and non-primitive data types.

* **primitive data type(call by value):**

Primitive values are data that are stored directly in a variable. These include numbers, booleans, strings, null, and undefined. When we assign a primitive value to a variable, a copy of that value is created and stored in memory. Any changes made to the variable do not affect the original value.

* **non-primitive data type(call by reference):**

Non-primitive data are more complex data types that are stored in memory and accessed through a reference. These include arrays, objects, and functions. When we assign a reference value to a variable, a reference to the original value is created and stored in memory. Any changes made to the variable affect the original value.

1. GitHub Page 網址（作業成品）

<https://chiehchunlin.github.io/remote-assignments/Week-2/Assignment-4/>

<https://chiehchunlin.github.io/remote-assignments/Week-2/Assignment-6/>

1. GitHub 網址（作業程式碼）

<https://github.com/ChiehChunLin/remote-assignments.git>