JS

{ JavaScript }

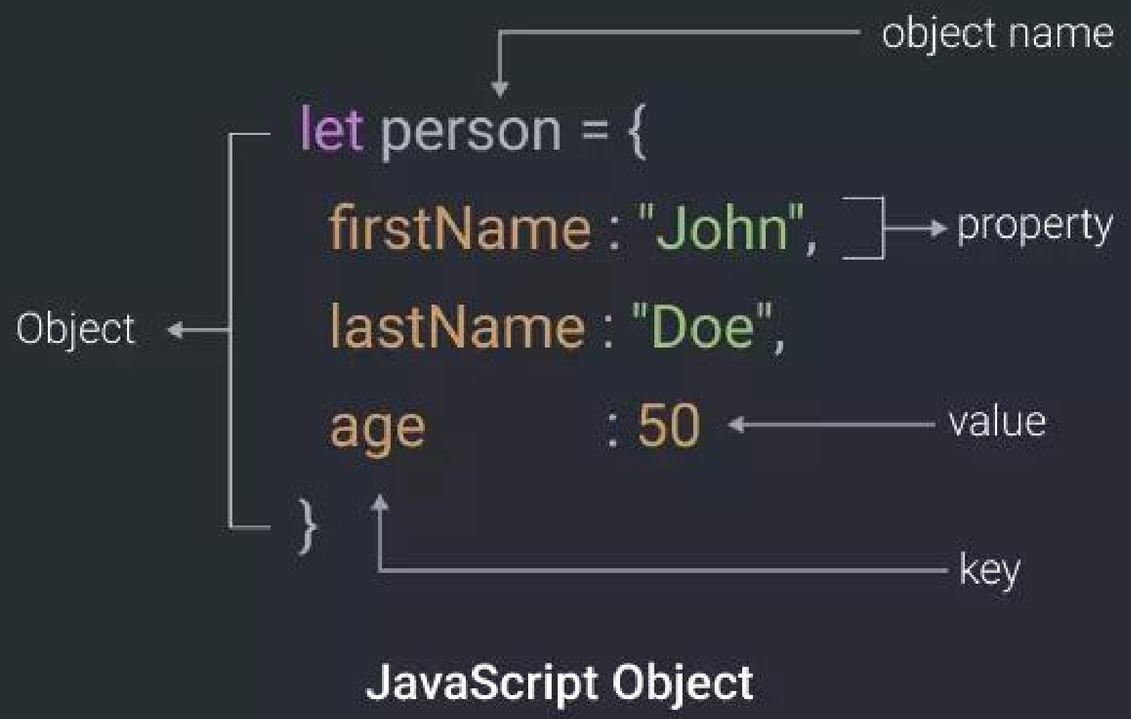
CRAFT KNOWLEDGE

Objectives

JavaScript Objects

- WHAT IS JS OBJECTS
- CREATEING JS OBJECTS
- ACCESSING OBJECTS PROPERTIY
- OBJECT METHODS
- THIS KEYWORD
- USING BUILT-IN METHODS
- ITERATING THROUGH OBJECTS

What is Javascript Objects



JavaScript Object

- Objects are variables too. But objects can contain many values.
- Object values are written as name : value pairs (name and value separated by a colon).
- An object is a collection of key/value pairs or properties.
- JavaScript objects are containers for named values, called properties and methods.

Creating JavaScript Objects

- JavaScript provides you with two ways to create an objects.

- Using an Object Literal

```
const Student = {
                    firstName:"John",
                    lastName:"kal",
                   age:50,
                    IsActive: true
                     };

 const Student = {};
          Student.firstName = "John";
          Student.lastName = "kal";
          Student.age = 50;
          Student.isActive = true;
```

Using the JavaScript Keyword new

```
var Student = new Object();
Student.first_name = "John";
Student.last_name = "kal";
Student.age = 50;
Student.isActive = true;
```

But there is no need to use new Object().

For readability, simplicity and execution speed, use the object literal method.

```
// 1. objectName.property

console.log(Student.first_name)  //John
console.log(Student.last_name)   //kal
console.log(Student.age)    // 50
console.log(Student.isActive) // true

//2. objectName["property"]

console.log(Student['first_name']) // John
console.log(Student['last_name'])  // kal
console.log(Student['age'])  // 50
console.log(Student['isActive']) // true
```

Accessing JavaScript object Properties

- Properties are the values associated with a JavaScript object.
- A JavaScript object is a collection of unordered properties.
- Properties can usually be changed, added, and deleted, but some are read only.

- we can access the properties of object by its key.
  objectName.property
  objectName["property"]

- **Deleting Properties**
- The delete keyword deletes a property from an object:

```
delete Student.rome;
delete Student['age'];
```

```
// 1.
var Student = {

    first_name: 'jhon',
    last_name:'kal',
    age:24,
    isActive: true,
    greet: function()
    {
        console.log('Hello');
    },
}
```

JavaScript Object Methods

- An object is a collection of key/value pairs or properties.
- When the value is a function, the property becomes a method. Typically, you use methods to describe the object behaviors.
- JavaScript methods are actions that can be performed on objects.
- Calling of A JavaScript method is a property containing a function definition.

Accessing Object Methods

- You access an object method with the following syntax:

  objectName.methodName()

  name = person.fullName();
- If you access the fullName property, without (), it will return the function definition:

  name = person.fullName;

- **Adding a Method to an Object**

  Adding a new method to an object is easy:

Example:-

```
person.name = function () {
   return this.firstName + " " + this.lastName;
};
```

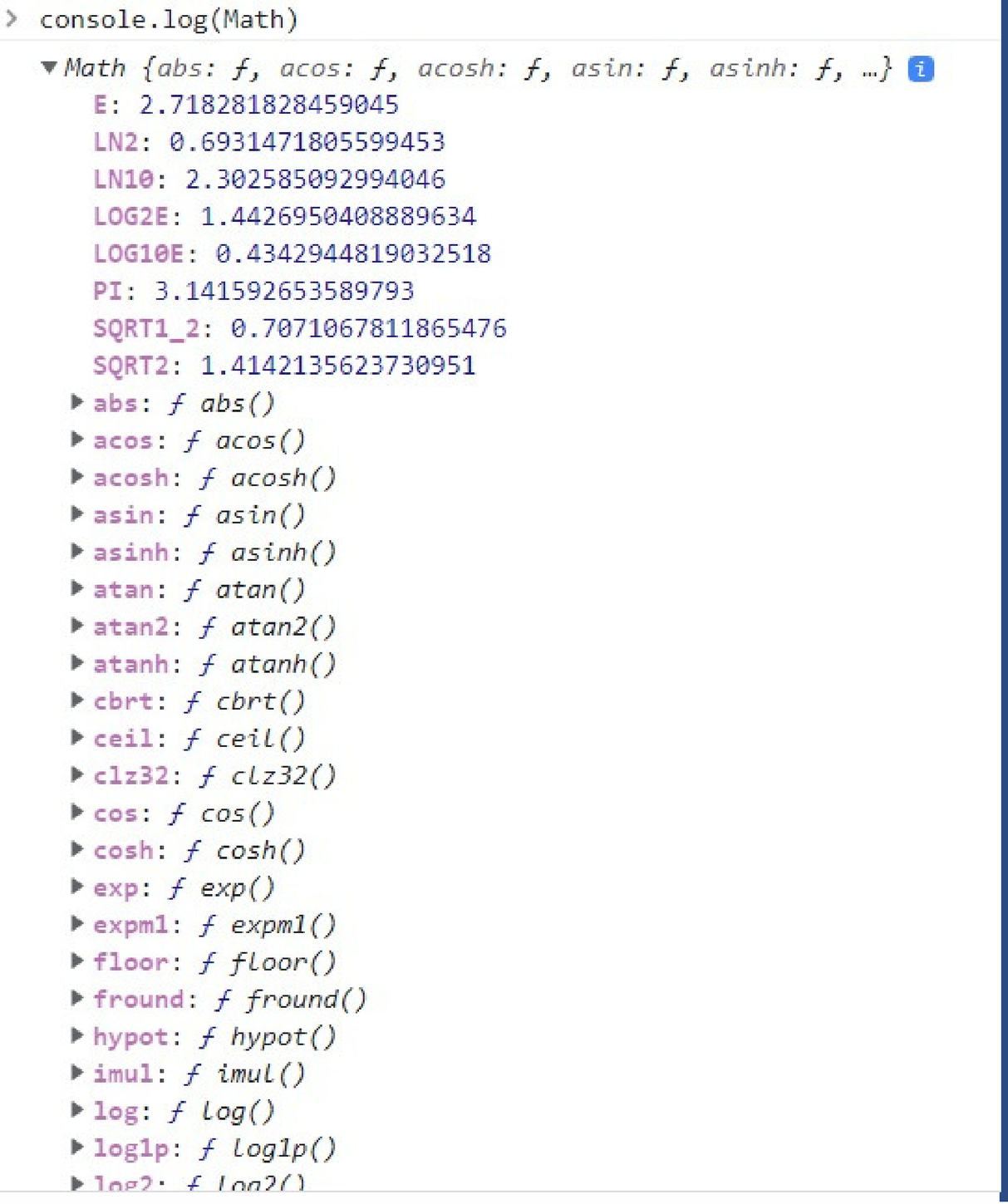
This keyword

- Methods need to access other properties of the object.

- For example, you may want to define a method that returns the full name of the Student object by concatenating the first name and last name.

- Inside a method, the this value references the object that invokes the method.
- Therefore, you can access a property using the this value as follows:

- **this**.propertyName



Using Built-In Methods

Using Built-In Methods

- There are lote of bulit in methods form Math object
  - floor()
  - round()
  - abs()
  - exp()
- This example uses the toUpperCase() method ofthe String object, to convert a text to uppercase:

```
let message = "Hello world!";
let x = message.toUpperCase();
```

Iterating Through Objects

```
for (let key in Student)
{
    console.log(key,':',Student[key])
}
```

1. .forEach() // don't work
2. for loop // don't work
3. for-of loop //don't work
4. for-in loop