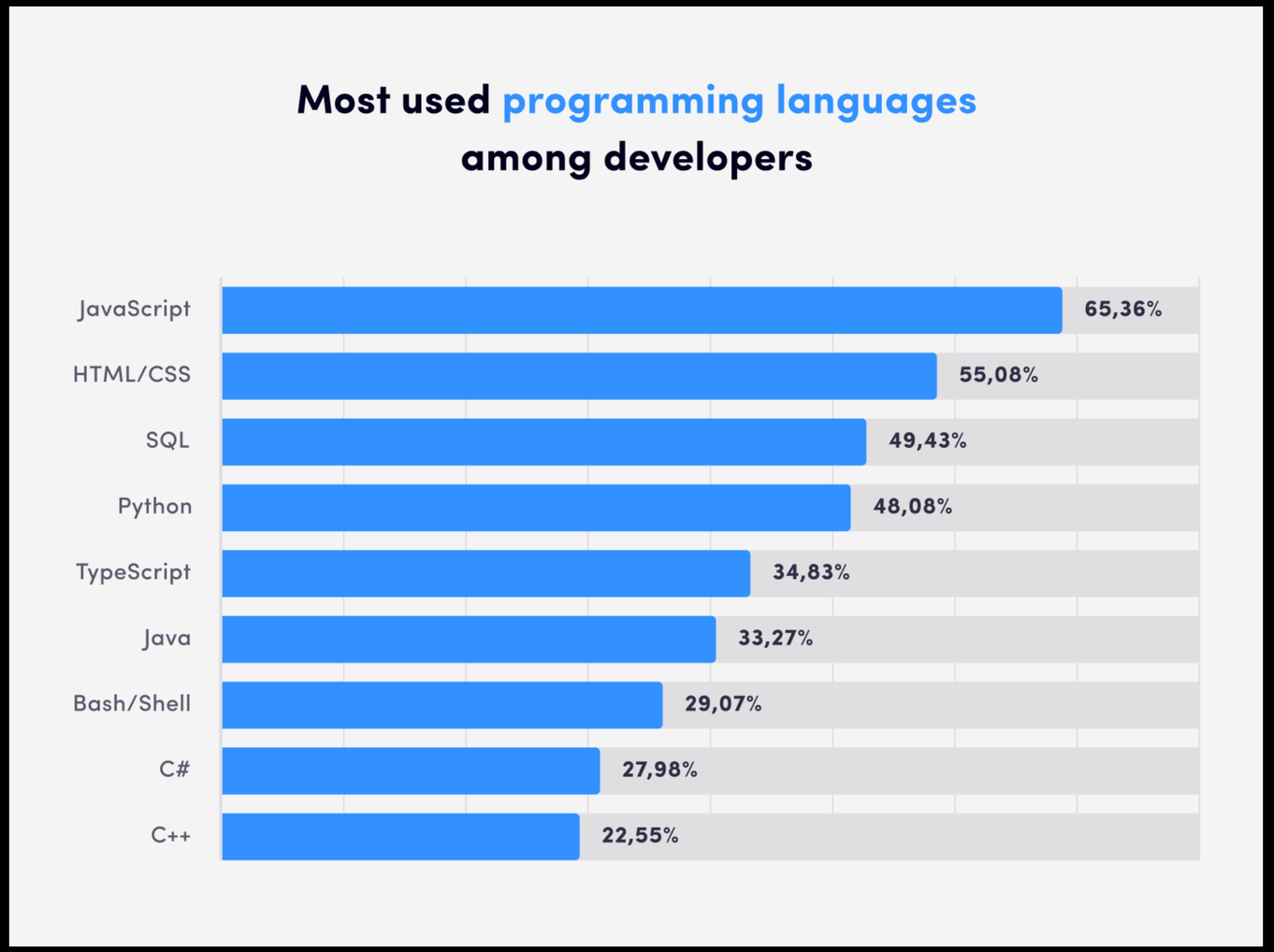
Javascript
Fundamentals

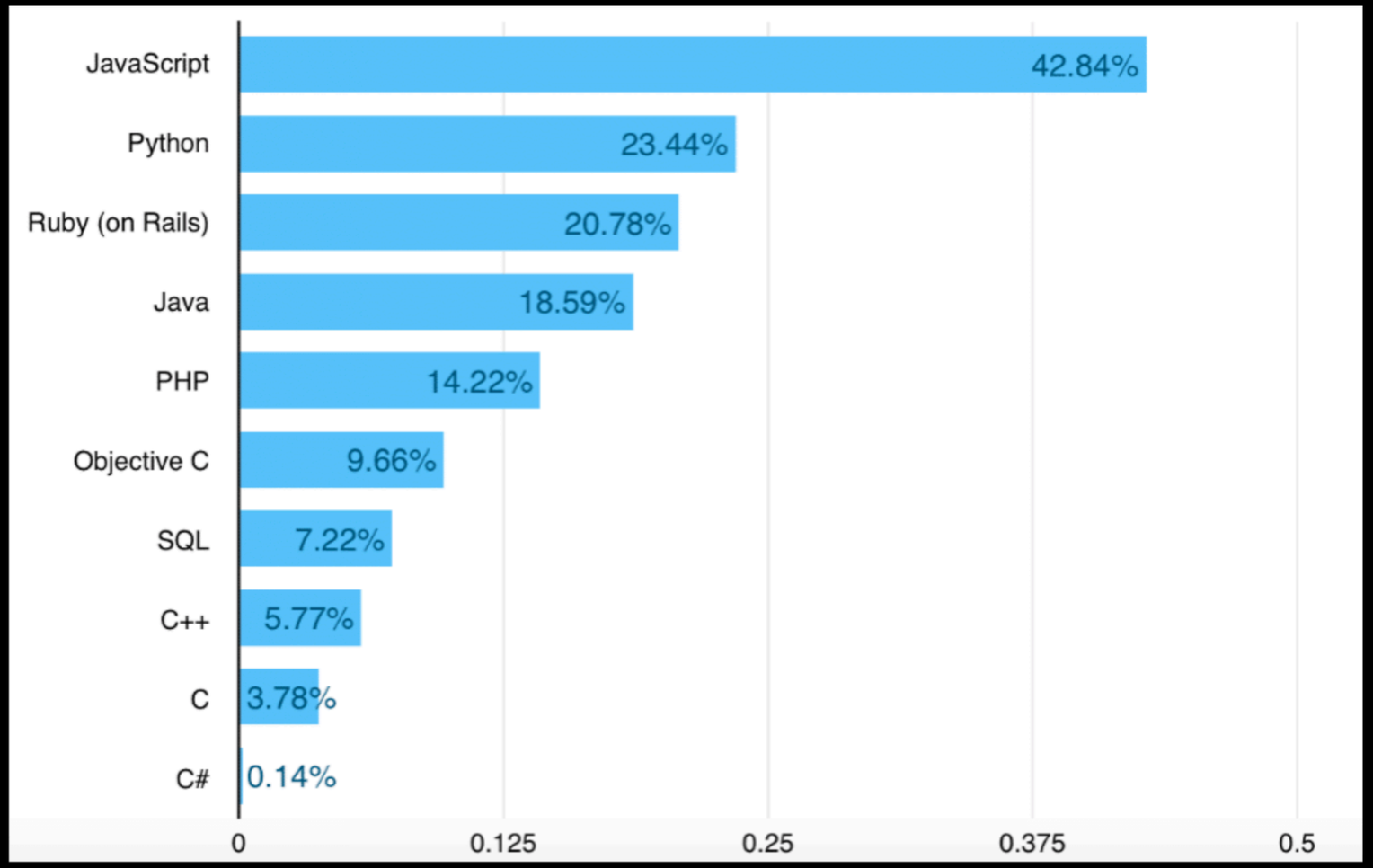
What is best programming language ?

For beginner

The market


Most used programming languages among developers
JavaScript
65,36%
HTML/CSS
55,08%
SQL
49,43%
Python
48,08%
TypeScript
34,83%
Java
33,27%
Bash/Shell
29,07%
C#
27,98%
C++
22,55%

The fastest learning language

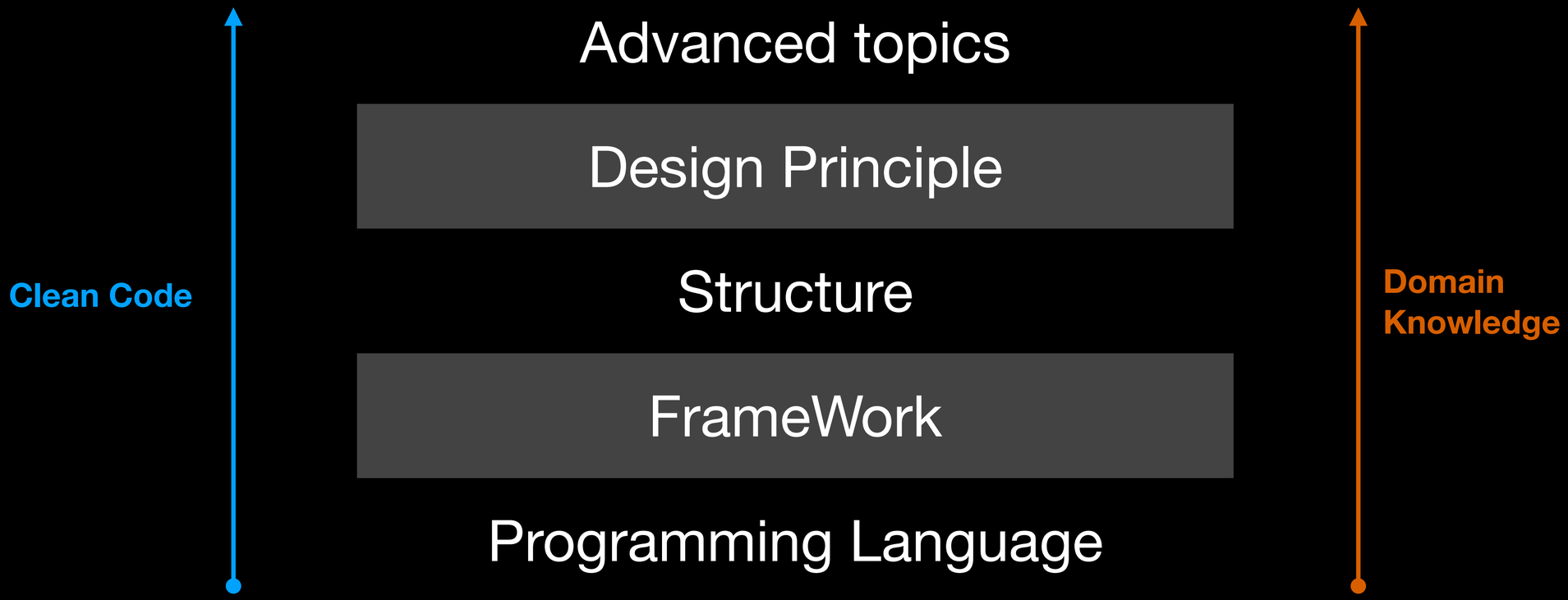


The right way to learn code

Mastering one first

Everything else become easier

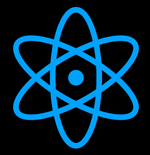
The right way to learn code



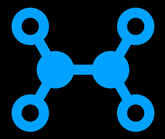
Learning source

- A-Z: MDN - Mozilla Developer Network
- Blog: Toidicodedao, fullsnack - huytd
- Advanced: How javascript work - session stack
- Books
  - Clean Code
  - Pragmatic Programmer

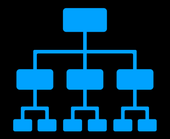
Javascript Course



Basics

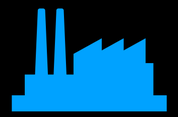


Functions



Module

Engine



Nodejs

Basics

Primitives	Operator	If else
Function	Array	Object

Primitives

Number	String	Boolean
Null	Undefined	Symbol

Primitives

```
// you can not change value of a primitive
let myName = "Quylaokame";

let shortName = myName.slice(6);

console.log(shortName) // "kame";

console.log(myName) // "Quylaokame";

// you just can assign your variable to another value
myName = "kame";
```

String and Number

```
let x = 5 + 7;
// 12
x = 5 + "7";
// 57
x = 5 - "x";
// NaN
x = null - 15;
// -15
x = undefined - 15;
// NaN
```

```
typeof NaN// number

// careful with js float
x = 0.1 + 0.2;
(x === 0.3) // false

// converted string to number
let string = “-012.45”;
let score = "1.7e19"
num = Number(string);
let num = +string;
```

undefined

```
let x;
// x is undefined
console.log(y);
//Uncaught ReferenceError: y is not defined

function add(x, y) {
    console.log(x, y)
    return x + y;
};
add(3);
// x = 3, y = undefined
typeof undefined; // "undefined";
```

null

Non-zero value

0

null

undefined

Falsy values javascript

```
// falsy value javascript

Boolean(null); // false

!!undefined // false

!!0 // false
```

```
// falsy value javascript

!!"" // false

!!"0" // true

!!NaN // false
```

if else

```
x = 0; // falsy values…
if (x) // false

// the most difficult bug in js
if (x = 10) {
    // it is true
}
if (x = 0) {

} // it is false
```

```
x = 10;

switch (x) {

  case "10": console.log(“Hello”);
  // don't log anything

  case 10: alert("Hello");
  // it works
```

Array access

```
var cars = ["Mazda", "Volvo", "BMW"];
var user = ["John", "Jim", "Jay"];

cars[0] // "Mazda";
cars[1] = "Toyota";

const [car1, car2, car3] = cars;
//car1: "Mazda", car2: “Toyota"
```

```
let arr = [0, 1, 2];
arr[4] = “4";
// [0, 1, 2, empty, 4]
console.log(arr[3]) // undefined

arr.length = 10;
// [0, 1, 2, empty, "new", empty × 5]
arr.length = 2;
console.log(arr) // [0, 1]
```

Array add, remove item

```
let letters = ["A","B","C"];
// add items to end of array
letters.push("D"); // 3
// return the length of fruits

// add items to start of array
letters.unshift("O"); // 4

// Remove an item from the end
let last = letters.pop(); // "D"

// Remove an item from beginning
let first = letters.shift(); // "O"
```

```
//Remove an item by index position
let secondItem = fruits.splice(1, 1);

delete letters[2];
console.log(letters);//  ["A",empty,"C"];

//Replace items by index
letters.splice(1, 2, "E", "I");// ["A","E","I"];

//empty array
letters.length = 0;
letters = [];
```

Array Looping

```
const numbers = [1, 4, 9, 16];
// for loop
for (let i = 0; i < numbers.length; i++) {
    console.log(numbers[i]);
}

const len = numbers.length;
for (let i = 0; i < len; i++) {
    console.log(numbers[i]);
}
// better is using foreach
numbers.forEach(num => console.log(num));
```

```
const doubles = numbers.map(function(num){
    return num * 2;
}); // return new Array [2, 8, 8, 32]

const odds = numbers.filter(function(item){
    return (item % 2) === 1;
}); // [ 16,4 ]

// sorting ascending
numbers.sort((a, b) => a - b);
// sorting decreasing
numbers.sort((a, b) => b - a);
```

Array Reduce

```
numbers = [2, 2, 3, 4];
const sum = numbers.reduce(function(total, num){
   return total + num;
});
// 11

let total = 0;
numbers.forEach(num => total += num);

let product = numbers.reduce(function (prev, current) {
    return prev * current;
});
// same as
product = numbers.reduce((prev, current) => prev * current);
```

Array common methods

```
let fruits = ['Apple', 'Banana'];
typeof fruits
// object

Array.isArray(fruits) // true

fruits.indexOf('Banana'); // 1
numbers.includes(4); // false

let clones = fruits.slice();
```

```
numbers = [1, 3, 5, 7, 9];
let squareNumber = numbers.find(function(num){
    return Number.isInteger(Math.sqrt(num));
});

let hasNegative = numbers.some(num => num < 0);

let areAllIntegers = numbers.every(num => num > 0);

let decreasingNumbers = numbers.reverse();
```

Object

```
const person = {
    firstName: "John",
    lastName: "Doe",
    age: 50,
    eyeColor: "blue"
};
function hello(person){
    const { firstName, lastName } = person;
    console.log("Hi " + firstName + " " + lastName);
}
```

Contructor Pattern

```
function Player(userName, score) {
    this.userName = userName;
    this.score = score;
}

const kame = new Player('quylaokame', 1000);
kame.avatar = “./database/avatar/kame";

//iterate over an object
for (let prop in kame) {
  console.log(kame[prop]);
}
```

Object common methods

```
Object.create(source);
// Creates a new object extends source

Object.assign(target, source)
// Copies all values from source to target

Object.keys(obj)
// Returns an array containing all properties in obj

Object.values()
// Returns an array containing all values in obj

Object.freeze()
// Freezes an object.
```

Assignment

```
# Exercise 1

1. Write a function to format money string:
10000000 => “10,000,000" || 123456 => “123,456" || 12000.02 => "12,000.02"

2. Write a function for format money in shorten :
1000 => 1K, 1123400000 => 1.12B , 1342222 => 1.34M

3. Write the function to count how many words appear in a string:
oneTwoThree => 3

4. Write the function get the get the Extension of file:
“image.png” => “png”. “Sound.mp3” => “mp3”
```

Assignment

```
# Exercise 2

1. Write the function to calculate the combination (Ckn)

2. Write the function to get a random integer between 2 numbers: min, max;

3. Write the function get a random element from an arrays.

3. Given two arrays of integers,
find which elements in the second array are missing from the first array.
```

Assignment

```
# Exercise 3: 2D Arrays

- We have a rectangle garden with 3 row and 5 column:

- each cell had a bomb or no bomb: ( 0: SAFE, 1: BOMB)

- problem: find all the safe way to go from the left to the right of the
  garden.

- examples:
Input: [ [0,1,1], [0,1,1], [0,1,1], [0,1,1], [0,0,1]]
Output : [ [0,0,0,0,0] , [0,0,0,0,1]]
```

Assignment

```
# Exercise 4: Complete your test

1. convert to Roman number: n integer < 1,000

2. print how to read number in vietnamese: n integer < 1,000,000
726503 : bảy mươi hai vạn sáu ngàn năm trăm linh ba.
```

How to ask

1. What you want to do?

2. What did you do, and what is your issue?

3. Capture of your code or the error.