JS

String Number Boolean Symbol Undefined Null

1.typeof typeof ( ) 7 number boolean symbol string object undefined

```
typeof ''; // string
typeof 1; // number
typeof Symbol(); // symbol
typeof true; //boolean
typeof undefined; //undefined
typeof null; //object
typeof [] ; //object
typeof new Function(); // function
typeof new Date(); //object
typeof new RegExp(); //object
```

- null
- function object
- null object
- function function

2 instanceof instanceof

```
instanceof (A,B) = {
    var L = A.__proto__;
    var R = B.prototype;
    if(L === R) {
        // A    __proto__    B
        return true;
    }
    return false;
}
//
[] instanceof Array; // true
{} instanceof Object;// true
new Date() instanceof Date;// true

function Person(){};
new Person() instanceof Person;

[] instanceof Object; // true
```

```
new Date() instanceof Object;// true
new Person instanceof Object;// true
```

3 constructor F JS F prototype prototype constructor

```
> function F(){}
< undefined
> F.prototype
< ▼Object {}
    ▶constructor: function F()
    ▶__proto__: Object
>
```

F

```
> ''.constructor == String
< true
> new Number(1).constructor == Number
< true
> true.constructor == Boolean
< true
> new Function().constructor == Function
< true
> new Date().constructor == Date
< true
> new Error().constructor == Error
< true
> [].constructor == Array
< true
> document.constructor == HTMLDocument
< true
> window.constructor == Window
< true
>
```

constructor

> 1. null undefined constructor > 2. constructor prototype constructor constructor Object ##### 4 toString toString() Object [[Class]] [object Xxx] Xxx

```
Object.prototype.toString.call('') ;   // [object String]
Object.prototype.toString.call(1) ;    // [object Number]
Object.prototype.toString.call(true) ; // [object Boolean]
Object.prototype.toString.call(Symbol()); //[object Symbol]
Object.prototype.toString.call(undefined) ; // [object Undefined]
Object.prototype.toString.call(null) ; // [object Null]
Object.prototype.toString.call(new Function()) ; // [object Function]
Object.prototype.toString.call(new Date()) ; // [object Date]
Object.prototype.toString.call([]) ; // [object Array]
Object.prototype.toString.call(new RegExp()) ; // [object RegExp]
Object.prototype.toString.call(new Error()) ; // [object Error]
```

```
Object.prototype.toString.call(document) ; // [object HTMLDocument]
Object.prototype.toString.call(window) ; //[object global] window     global
```

Vue3 watchEffect

watch

- watchEffect watch (v3)
- watch watchEffect
- watchEffect computed watch #### import require ##### import ES6 module 1. 2. Module 3. Module API export 4.import module ##### require
- CommonJS ES6
- CommonJS ES6 import
- CommonJS #### script

1. async: js load load js defer: js js async integrity SRI Subresource Integrity CDN
2. crossorigin CORS CORS crossorigin=“anonymous” crossorigin=“use-credentials” script crossorigin
   - crossorigin script request origin cors response header ‘Access-Control-Allow-Origin’ cors script window.onerror error.message
   - crossorigin anonymous use-credentials crossorigin anonymous
   - anonymous cookie cookie fecth credentials: ‘same-origin’
   - use-credentials cookie fetch credentials: ‘include’ response header ’Access-Control-Allow-Credentials’ = true cors
3. language ”JavaScript” “JavaScript 1.2” ”VBScript”
4. src charset src
5. type language MIME ”text/javascript” ”text/javascript” ”text/ecmascript” JavaScript MIME ”application/x-javascript” type- IE ”application/javascript” ”application/ecmascript” module vite ES6 import export

http2 TCP

HTTP/2 HTTP/2 TCP HTTP/2

1. TCP
2. 3. 31bit 0

1. frame stream

frame HTTP/2 HTTP/1.x HTTP/1.x HTTP/1.x length type flags stream identifier frame payload type HTTP/2 10 HEADERS frame DATA frame

`PRIORITY` `RST_STREAM` `SETTINGS` `PUSH_PROMISE` `PING` `RTT` `GOAWAY` `WINDOW_UPDATE` `CONTINUATION`

HTTP 2.0 header frame data frame

stream ID HTTP/2 - (stream)

-
-
-
- ,
-
- HEADERS DATA
- ID
- HTTP/2 frame stream
-
- HTTP2 HTTP1.x - HTTP1.x TCP Tcp HTTP

useCallback useMemo

-	useCallBack	useMemo
	( )	
	props	
	useCallBack	useMemo

```
// useCallBack
import React, { useCallback, useEffect, useState } from "react";
import "./styles.css";

export default function App() {
  const [count, setCount] = useState(0);

  //   useCallBack
  const handleCountAddByCallBack = useCallback(() => {
    setCount((count) => count + 1);
  }, []);

  //     count
```

```
  const handleCountAdd = () => {
    setCount((count) => count + 1);
  };

  return (
    <div className="App">
      <h3>CountAddByChild1: {count}</h3>
      <Child1 addByCallBack={handleCountAddByCallBack} add={handleCountAdd} />
    </div>
  );
}

const Child1 = React.memo(function (props) {
  const { add, addByCallBack } = props;

  //        memo               useEffect
  useEffect(() => {
    console.log("Child1----addFcUpdate", props);
  }, [add]);

  //   memo          useEffect
  useEffect(() => {
    console.log("Child1----addByCallBackFcUpdate", props);
  }, [addByCallBack]);

  return (
    <div>
      <button onClick={props.add}>+1</button>
      <br />
      <button onClick={props.addByCallBack}>+1(addByCallBack)</button>
    </div>
  );
});
// useMemo
import { useState, useMemo } from "react";
import "./styles.css";

export default function App() {
  const [count, setCount] = useState(0);
  const [total, setTotal] = useState(0);

  //    useMemo    total, countToString
  const countToString = (() => {
    console.log("countToString   ");
    return count.toString();
  })();
```

```
  //    useMemo,    total
  const totalToStringByMemo = useMemo(() => {
    console.log("totalToStringByMemo   ");
    return total + "";
  }, [total]);

  return (
    <div className="App">
      <h3>countToString: {countToString}</h3>
      <h3>countToString: {totalToStringByMemo}</h3>
      <button
        onClick={() => {
          setCount((count) => count + 1);
        }}
      >
        Add Count
      </button>
      <br />
      <button
        onClick={() => {
          setTotal((total) => total + 1);
        }}
      >
        Add Total
      </button>
    </div>
  );
}
```

1. useCallBack React.memo 2. useMemo

eventLoop

& EventLoop set Queue grab dequeuing ##### setTimeout setInterval setImmediate (Node) requestAnimationFrame ( ) I/O UI rendering ( ) ##### process.nextTick (Node) Promise Object.observe MutationObserver ##### eventLoop main thread task 1 task 2 1 3 eventLoop 4 mainThread stack task evenLoop eventLoop callback callback eve

ES6 Async/Await ,Generator Promise

async/await

1. async await
2. wait async
3. await ##### async Generator Generator next() dva Generator async Generator ,async Generator 4 :
4. : async , asyncReadFile()
5. : async ,await
6. : yield , Thunk Promise , async await , Promise
7. Promise , then

async Generator await Promise async then