

1.下图代码执行后输出的结果是? (选择一项)

```
<script>
  function foo(n){
    const f = () => {
      const [arg1] = arguments;
      return arg1 + n;
    };
    return f();
  }

  let res = foo(2);
  console.log(res);
</script>
```

- A.2
- B.NaN
- C.4
- D.报错

2,下图代码执行后输出的结果是? (选择一项)

```
<script>
  const foo = ({ x = 1, y = 2, z = 3 }) => {
    console.log(x, y, z);
  };

  foo({
    z: 2,
    y: 1,
    x: 0,
  });
</script>
```

- A.1 2 3
- B.2 1 0
- C.undefined 1 undefined
- D.0 1 2

3.下列代码执行结果是？（选择一项）

```
<script>
  const map = new Map();
  const s = new Set([1,NaN,2,1, "NaN", NaN]);
  map.set([1],"imooc")
  const m1 = map.get([1]);
  console.log(m1)
  console.log(s.size)
</script>
```

A.imooc 4

B.imooc 5

C.undefined 5

D.undefined 4

4.下列选项中，不可以使用for...of循环的是？（选择一项）

A.let arr = ['y', 'u', 'n', 'm', 'u']

B.let obj = {name: "xm", age: 18}

C.let str = 'yunmudsb'

D.let list = document.querySelectorAll('li')

5.关于promise对象的状态，下列描述错误的是？（选择一项）

A.pending初始状态可以变成fulfilled成功状态

B.pending初始状态可以变成rejected失败状态

C.rejected失败状态可以变成fulfilled成功状态

D.fulfilled成功状态不可以变成rejected失败状态

6.下图所示代码，运行结果是？（选择一项）

```
<script>
  Promise.resolve(1)
    .then((x) => x + 1)
    .then((x) => {
      throw new Error("err ");
    })
    .catch(() => 1)
    .then((x) => x + 1)
    .then((x) => console.log(x))
    .catch((x) => console.log(error));
</script>
```

A.3 err

B.2 err

C.3

D.2

7.下图所示代码中，函数fn内部返回了一个Promise实例对象，如果要实现下图中的输出结果，并且是每隔一秒输出一次数值1，选项中代码书写正确的是？（选择一项）

```
<script>
  const fn = () => {
    return new Promise((resolve, reject) => {
      setTimeout(() => {
        resolve(1);
      }, 1000);
    });
  };
</script>
```

A.

```
fn().then((data) => {
  console.log(data);
})
  .then((data) => {
    console.log(data);
  })
  .then((data) => {
    console.log(data);
  });
```

B.

```
fn().then((data) => {  
  console.log(data);  
  return data;  
})  
.then((data) => {  
  console.log(data);  
  return data;  
})  
.then((data) => {  
  console.log(data);  
});
```

C.

```
fn().then((data) => {  
  console.log(data);  
  fn();  
})  
.then((data) => {  
  console.log(data);  
  fn();  
})  
.then((data) => {  
  console.log(data);  
});
```

D.

```
fn().then((data) => {  
  console.log(data);  
  return fn();  
})  
.then((data) => {  
  console.log(data);  
  return fn();  
})  
.then((data) => {  
  console.log(data);  
});
```

8.下列所示代码，运行结果是？（选择一项）


```
<script>
  const age = 17;
  class Person {
    constructor(age) {
      this.age = age;
    }
    age = 18;
    static age = 19;
    say = () => {
      console.log(this.age);
    };
  }
  let p1 = new Person(20);
  p1.say();
</script>
```

A.17

B.18

C.19

D.20

9. 下列代码中，书写格式正确的选项是？（选择两项）

A.

```
<script>
  export default 1;
  export const add = (a, b) => a + b;
</script>
```

B.

```
const m = 1
export m
```

C.

```
const age = 1
const sex = 'male'
export default age
export default sex
```

D.

```
import * as user from './index.js'
```

10.下图代码运行结果是？（选择一项）

```
<script>  
    const arr = [...[... '...']]  
    console.log(arr.length)  
</script>
```

- A.0
- B.1
- C.3
- D.报错

11.下图所示代码，运行后输出结果是？（选择一项）

```
<script>
  let a = 1;
  const foo = () => {
    return (obj = {
      a: 2,
      fn() {
        let a = 3;
        setTimeout(() => {
          console.log(this.a);
        });
      },
    });
  };
  const result = foo();
  result.fn();
</script>
```

A.0

B.1

C.3

D.报错

12.使用webpack对项目进行打包，下列说法正确的是？（选择两项）

- A.**webpack可以自定义同时配置多入口和多出口文件
- B.**html、css以及js中的图片可以使用file-loader处理
- C.**在js中使用require方式引入的css文件可以使用css-loader转换
- D.**html-webpack-plugin插件只能创建一个html入口文件

13.根据如下代码，下列说法错误的是？（选择两项）

```
<script>
{
  let a = 1;
  {
    let a = 2;
    console.log(a); //第一次输出
  }
  console.log(a); //第二次输出a
}

{
  var b = 1;
  {
    var b = 2;
    console.log(b); //第一次输出b
  }
  console.log(b); //第二次输出b
}
</script>
```

- A.第一次输出a的值为2，第二次输出a的值也为2
- B.第一次输出a的值为2，第二次输出a的值为1
- C.第一次输出b的值为2，第二次输出b的值也为2
- D.第一次输出b的值为1，第二次输出b的值为2

14.下列关于常量与变量说法错误的是？（选择两项）

- A.变量是可以改变的
- B.常量声明必须赋值
- C.变量声明必须赋值
- D.常量可以重复声明

15.观察如下代码，最后输出的结果是？（选择一项）

```
<script>
  function bar(x = y, y = 2) {
    return [x, y];
  }
  console.log(bar());
</script>
```

- A. [2 2]
- B.[y 2]
- C.报错
- D.[x y]

16.如下图，是需要匹配的数组，下列哪个选项不能匹配到'f'？（选择一项）

```
const arr = [ 'a', 'b', [ 'c', 'd', [ 'e', 'f', 'g' ] ] ];
```

A.

```
let [ , , [ , , [ , f , ] ] ] = arr;
```


B.

```
let [, , [f]] = [ 'c', 'd', [ 'e',  
'f', 'g' ]];
```

C.

```
let [ , , [ , , [f]] ] = arr;
```

D.

```
let [ , , [ , f , ] ] = arr;
```

17.根据如下代码，哪项可以匹配到“前端进阶：响应式开发与常用框架”与“H5&JS进阶与组件化网页开发？”（选择一项）

```
const player = {
  classify: "前端",
  source: "潭州edu",
  Course: [
    {
      CourseName: "前端小白入门",
      price: 499,
    },
    [
      {
        CourseName: "前端进阶:响应式开发与常用框架",
        price: 599,
      },
    ],
    {
      CourseName: "H5&JS进阶与组件化网页开发",
      price: 699,
    },
  ],
};
```

A.

```
const { Course:[ , [{CourseName}] ,
{CourseName :Name}]}= player;
```

B.

```
const{ Course:[[{CourseName}],
{CourseName:Name}]} = player;
```

C.

```
const { Course:[ , [{CourseName}],  
{courseName}]} = player;
```

D.

```
const { , , [, [{CourseName}],  
{CourseName:Name}]} = player;
```

18.观察代码，下列输出结果正确的是？（选择一项）

```
<script>  
  const target = { a: { b: "c", d: "e" } };  
  const source = { a: { b: "hello" } };  
  const obj = Object.assign(target, source);  
  console.log(obj);  
</script>
```

A. { b: 'hello', d: 'e' }

B. { a: { b: 'c', d: 'e' } }

C. { a: { b: 'hello', d: 'e' } }

D.{ a: { b: 'hello' } }

19.学习了类表达式，使用类的表达式的方式书写如下代码，哪一项不能正确输出“云牧最帅”？（选择两项）

A.

```
let person = class {  
  constructor() {  
    console.log("云牧最帅");  
  }  
};  
new p();
```

B.

```
let person = class {  
  constructor() {  
    p.imo = "云牧最帅";  
    console.log(p.imo);  
  }  
};  
new person();
```

C.

```
let person = class {  
  constructor() {  
    console.log("云牧最帅");  
  }  
};  
new person();
```

D.

```
let person = new (class {  
  constructor() {  
    console.log("云牧最帅 ");  
  }  
})();
```

20.观察下如下代码，描述正确的选项是？（选择两项）

```
class Person {  
  constructor(x, y) {  
    this.x = x;  
    this.y = y;  
  }  
  eat() {  
    alert("前端");  
  }  
}  
  
class ColorPoint extends Person {  
  constructor(x, y) {  
    super(x, y);  
    this.x = "ES6";  
  }  
}  
  
let col = new ColorPoint("js", "javascript");
```

A.

console.log(col.x); ==> 输出js

B.

col.eat(); ==> 输出前端

C.

console.log(col.x); ==> 输出ES6

D.

```
col.eat(); ==> 报错
```