

# 预解析

提问：以下代码的输出分别是多少？

```
<script>
    console.log(a);    //输出?
</script>

<script>
    console.log(num);  //输出?
    var num = 1;
</script>

<script>
    var fn1 = function(){
        console.log(111);
    }
    fn1();              //输出?
</script>

<script>
    fn2();              //输出?
    var fn2 = function(){
        console.log(222);
    }
    // fn2();
</script>
```

## 变量预解析(变量提升)

JavaScript 代码是由浏览器中的 JavaScript 解析器来执行的。JavaScript 解析器在运行 JavaScript 代码的时候分为两步：预解析 和 代码执行

一、js引擎运行js分为两步：预解析 和 代码执行

1. 预解析：js引擎会把js里面所有的 var 还有 function 提升到 当前作用域 的前面
2. 代码执行：按照代码书写的顺序从上往下执行

二、预解析分为 变量预解析（变量提升）和 函数预解析（函数提升）

1. 变量提升：就是把所有的变量声明提升到**当前的作用域**最前面，但是不提升赋值操作。

**例子1：变量提升**

```

<script>
    // 变量提升，把所有变量声明到当前作用域的最前面，但是不提升赋值操作。
    console.log(num);    //输出?
    var num = 1;

    // 以上代码相当于执行了以下代码
    var a;
    console.log(a);
    a = 10;

</script>

```

## 例子2: 函数表达式 (匿名函数) 提升

```

<script>
    // fn2();
    // var fn2 = function demo() {
    //     console.log(222);
    // }

    var fn3 = function() {
        console.log(333);
    }
    fn3();
</script>

```

2. 函数提升: 函数的声明会被提升到**当前作用域**的最前面，但是不会调用函数。

## 例子1: 自定义函数 (命名函数) 提升

```

<script>
    fn1();           //输出?
    function fn1(){
        console.log(111);
    }
</script>

```

## 案例1: 以下代码输出结果是多少?

```

<script>
    var num = 60;
    fun();
    function fun(){
        console.log(num);
        var num = 20;
    }
</script>

```

上述代码的执行操作    ↑

```
var num;
function fun() {
    var num;
    console.log(num);
    num = 20;
}
num = 10;
fun();
```

**案例2：以下代码输出结果是多少？**

```
<script>
var num = 10;
function fn() {
    console.log(num);
    var num = 20;
    console.log(num);
}
fn();
</script>
```

上述代码的执行操作    ↑

```
<script>
var num;
function fn(){
    var num;
    console.log(num);
    num = 20;
    console.log(num);
}
num = 10;
fn();
</script>
```

**案例3：以下代码输出结果是多少？**

```
<script>
var a = 18;
fn1();
function fn1(){
    var b = 9;
    console.log(a);
    console.log(b);
    var a = '123';
}
</script>
```

#### 案例4：以下代码输出结果是多少？

```
<script>
  f1();
  console.log(c);
  console.log(b);
  console.log(a);
  function f1() {
    var a = b = c = 9;
    // 相当于 var a = 9; b = 9; c = 9; b和c的前面没有var声明,当全局变量看
    // 集体声明 var a = 9, b = 9, c = 9;
    console.log(a);
    console.log(b);
    console.log(c);
  }
</script>
```

上述代码的执行操作 ↑

```
<script>
  function f1() {
    var a;
    a = b = c = 9;
    console.log(a); //9
    console.log(b); //9
    console.log(c); //9
  }
  f1();
  console.log(c); //9
  console.log(b); //9
  console.log(a); //报错 a是局部变量

</script>
```