

# 分支结构

---

- 我们的 js 代码都是顺序执行的（从上到下）
- 逻辑分支就是根据我们设定好的条件来决定要不要执行某些代码

## IF 条件分支结构

---

### if 语句

- 通过一个 if 语句来决定代码执行与否
- 语法：`if (条件) { 要执行的代码 }`
- 通过 `()` 里面的条件是否成立来决定 `{ }` 里面的代码是否执行

```
1. // 条件为 true 的时候执行 {} 里面的代码
2. if (true) {
3.   alert('因为条件是 true, 我会执行')
4. }
5.
6. // 条件为 false 的时候不执行 {} 里面的代码
7. if (false) {
8.   alert('因为条件是 false, 我不会执行')
9. }
```

### if else 语句

- 通过 if 条件来决定，执行哪一个 `{ }` 里面的代码
- 语法：`if (条件) { 条件为 true 的时候执行 } else { 条件为 false 的时候执行 }`
- 两个 `{ }` 内的代码一定有一个会执行

```
1. // 条件为 true 的时候，会执行 if 后面的 {}
2. if (true) {
3.   alert('因为条件是 true, 我会执行')
4. } else {
5.   alert('因为条件是 true, 我不会执行')
6. }
7.
```

```
..
8. // 条件为 false 的时候，会执行 else 后面的 {}
9. if (false) {
10.   alert('因为条件为 false，我不会执行')
11. } else {
12.   alert('因为条件为 false，我会执行')
13. }
```

## if else if ... 语句

- 可以通过 if 和 else if 来设置多个条件进行判断
- 语法：`if (条件1) { 条件1为 true 的时候执行 } else if (条件2) { 条件2为 true 的时候执行 }`
- 会从头开始依次判断条件
  - 如果第一个条件为 true 了，那么就会执行后面的 `{ }` 里面的内容
  - 如果第一个条件为 false，那么就会判断第二个条件，依次类推
- 多个 `{ }`，只会有一个被执行，一旦有一个条件为 true 了，后面的就不在判断了

```
1. // 第一个条件为 true，第二个条件为 false，最终会打印 “我是代码段1”
2. if (true) {
3.   alert('我是代码段1')
4. } else if (false) {
5.   alert('我是代码段2')
6. }
7.
8. // 第一个条件为 true，第二个条件为 true，最终会打印 “我是代码段1”
9. // 因为只要前面有一个条件满足了，就不会继续判断了
10. if (true) {
11.   alert('我是代码段1')
12. } else if (true) {
13.   alert('我是代码段2')
14. }
15.
16. // 第一个条件为 false，第二个条件为 true，最终会打印 “我是代码段2”
17. // 只有前一个条件为 false 的时候才会继续向后判断
18. if (false) {
19.   alert('我是代码段1')
20. } else if (true) {
21.   alert('我是代码段2')
```

```
22. }
23.
24. // 第一个条件为 false，第二个条件为 false，最终什么也不会发生
25. // 因为当所有条件都为 false 的时候，两个 {} 里面的代码都不会执行
26. if (false) {
27.     alert('我是代码段1')
28. } else if (false) {
29.     alert('我是代码段2')
30. }
```

## if else if ... else 语句

- 和之前的 `if else if ...` 基本一致，只不过是在所有条件都不满足的时候，执行最后 `else` 后面的 `{}`

```
1. // 第一个条件为 false，第二个条件为 false，最终会打印 “我是代码段3”
2. // 只有前面所有的条件都不满足的时候会执行 else 后面的 {} 里面的代码
3. // 只要前面有一个条件满足了，那么后面的就都不会执行了
4. if (false) {
5.     alert('我是代码段1')
6. } else if (false) {
7.     alert('我是代码段2')
8. } else {
9.     alert('我是代码段3')
10. }
```

## SWITCH 条件分支结构

---

- 也是条件判断语句的一种
- 是对于某一个变量的判断
- 语法：

```
1. switch (要判断的变量) {
2.     case 情况1:
3.         情况1要执行的代码
4.         break
5.     case 情况2:
6.         情况2要执行的代码
7.         break
```

```
8.   case 情况3:
9.       情况3要执行的代码
10.  break
11.  default:
12.      上述情况都不满足的时候执行的代码
13.  }
```

- 要判断某一个变量 等于 某一个值得时候使用

- 例子：根据变量给出的数字显示是星期几

```
1.  var week = 1
2.  switch (week) {
3.      case 1:
4.          alert('星期一')
5.          break
6.      case 2:
7.          alert('星期二')
8.          break
9.      case 3:
10.         alert('星期三')
11.         break
12.      case 4:
13.         alert('星期四')
14.         break
15.      case 5:
16.         alert('星期五')
17.         break
18.      case 6:
19.         alert('星期六')
20.         break
21.      case 7:
22.         alert('星期日')
23.         break
24.      default:
25.         alert('请输入一个 1 ~ 7 之间的数字')
26.  }
```

## 三元运算（扩展）

---

- 三元运算，就是用 两个符号 组成一个语句

- 三元运算只是对 `if else` 语句的一个简写形式

- 语法: `条件 ? 条件为 true 的时候执行 : 条件为 false 的时候执行`

1. `var age = 18;`
2. `age >= 18 ? alert('已经成年') : alert('没有成年')`