Day20笔记

|  |
| --- |
| **1. 操作符**  1.1 算术运算符   * **加法 (+)**: 数字相加，字符串拼接。   var num = 1 + 2; // 3  var str = "Hello" + " " + "World"; // "Hello World"   * **减法 (-)**: 数字相减，可以将字符串转换成数字后相减。   var num = 4 - 3; // 1  var num = "8" - "4"; // 4   * **乘法 (\*)**: 数字相乘，可以将字符串转换成数字后相乘。   var num = 4 \* 3; // 12   * **除法 (/)**: 数字相除，可以将字符串转换成数字后相除。   var num = 6 / 3; // 2   * **取余 (%)**: 取余数操作。   var num = 7 % 3; // 1  1.2 一元运算符   * **正号 (+)**: 将操作数转换成数字。   var num = +"123"; // 123   * **负号 (-)**: 将操作数转换成负数。   var num = -"123"; // -123   * **逻辑取反 (!)**: 将布尔值取反。   var bool = !true; // false   * **递增 (++)**: 变量自增1。   var num = 1;  num++; // 2   * **递减 (--)**: 变量自减1。   var num = 1;  num--; // 0   * **删除 (delete)**: 删除对象或数组的属性。   var obj = { name: 'John', age: 30 };  delete obj.age; // { name: 'John' }   * **类型判断 (typeof)**: 返回操作数的类型。   var type = typeof 123; // "number"  **2. 赋值运算符**   * **赋值 (=)**: 赋值操作。   var x = 5;   * **加等 (+=)**: 相加后赋值。   var x = 5;  x += 2; // 7   * **减等 (-=)**: 相减后赋值。   var x = 5;  x -= 2; // 3   * **乘等 (\*=)**: 相乘后赋值。   var x = 5;  x \*= 2; // 10   * **除等 (/=)**: 相除后赋值。   var x = 5;  x /= 2; // 2.5   * **取余等 (%=)**: 取余后赋值。   var x = 5;  x %= 2; // 1  **3. 比较运算符**   * **等于 (==)**: 仅值相等即可。   var result = (5 == "5"); // true   * **全等 (===)**: 值和类型都要相等。   var result = (5 === "5"); // false   * **不等于 (!=)**: 值不等即可。   var result = (5 != "5"); // false   * **不全等于 (!==)**: 值或类型不等。   var result = (5 !== "5"); // true   * **大于 (>)**: 大于。   var result = (5 > 3); // true   * **小于 (<)**: 小于。   var result = (5 < 3); // false   * **大于等于 (>=)**: 大于或等于。   var result = (5 >= 3); // true   * **小于等于 (<=)**: 小于或等于。   var result = (5 <= 3); // false  **4. 隐式转换**   * **ToPrimitive**: 将值转换为原始值。 * **ToNumber**: 将值转换为数字。 * **ToString**: 将值转换为字符串。   **隐式转换规则**   1. 字符串与数字相加时，数字会被转换为字符串。   var result = 1 + "2"; // "12"   1. 字符串与数字相减时，字符串会被转换为数字。   var result = "5" - 3; // 2  **5. 流程控制语句**   1. **if 语句**: 条件判断。   if (condition) {  // TODO  }   1. **switch 语句**: 多条件判断。   switch (expression) {  case value1:  // TODO  break;  case value2:  // TODO  break;  default:  // TODO  }   1. **for 循环**: 计数器控制的循环。   for (var i = 0; i < 5; i++) {  // TODO  }   1. **while 循环**: 条件控制的循环。   j  while (condition) {  // TODO  }   1. **do...while 循环**: 至少执行一次的循环。   do {  // TODO  } while (condition);  **使用注意**   * 尽量使用严格比较运算符（===），避免隐式转换带来的困惑。 * 注意类型转换的规则和顺序，避免意外的结果。   日期2024.7.22 |