

# Условные операторы

JavaScript

# Конструкция IF

- Ключевое слово `if`
- В скобках указывается `условие`
- Тело конструкции обособляется `{ }`
- Если условие истинно, то выполняется код из тела конструкции, если ложно, то конструкция пропускается
- ```
if (a > 0) {  
    alert('Число положительное');  
}
```

# Преобразование условия

- Оператор `if (...)` вычисляет и преобразует выражение в скобках к логическому типу
- В логическом контексте
  - `0`, `""`, `null`, `undefined` и `NaN` – `false`
  - Остальные значения — `true`

# else & else if

- **else if** также как и **if** имеет условие
- проверка условия для **else if** производится только в том случае, если условие для **if** оказалось ложным
- **else** не имеет условия и выполняется в случае если ни одно из предыдущих условий не было истинным
- ```
if (a > 0) { alert("Число положительное"); }  
else if (a < 0) { alert("Число отрицательное"); }  
else {alert("Число нулевое"); }
```

# Тернарный оператор ?:

- Представляет из себя укороченную форму записи конструкции if-else
- Состоит из 3-х частей
- условие ? значение1 : значение2
- Проверяется условие, если оно верно — возвращается значение1, иначе — значение2
- `access = age > 14 ? true : false;`

# Конструкция switch-case

- ```
switch(x) {  
    case 'value1': ...; [break];  
    case 'value2': ...; [break];  
    default: ...  
}
```
- Проверяем `x===value1`, затем `x===value2` и тд
- Если true, то выполняется соответствующий case *до* ближайшего *break*
- Если ни один case не совпал — выполняется default

# Пример

- В системе есть разграничения по правам, где 1 – разработчик, 2 – администратор, 3 – пользователь
- ```
a = 3;
switch (a) {
  case 1:
    console.info("Разработчик");
    break;
  case 2:
    console.info("Администратор");
    break;
  case 3:
    console.info("Пользователь");
    break;
  default:
    console.info("Не верный уровень доступа");
}
```