**УСЛОВНЫЕ КОНСТРУКЦИИ**

Конструкция if … else

Условные операторы контролируют поведение JavaScript приложения путем определения необходимости выполнения той или иной части кода.  
Базовый синтаксис конструкции if:

if (condition) {

statement

}

Результатом выполнения условия (condition) всегда будет логическое значение true или false.

if (25 > 5) {

console.log('Condition: true.');

}

Код внутри конструкции if будет выполнен только в том случае, если условие, заключенное в круглые скобки, будет равно true.  
Ключевое слово if дает знать JavaScript о том, что дальше следует условная конструкция, которую необходимо проверить. В том случае, если условие, заключенное в круглые скобки, равно true, то код, который находится между фигурными скобками выполнится. Если же результат будет противоположным (false), JavaScript пропустит код находящийся в конструкции if и продолжит выполнение кода вне конструкции.

Конструкция if…else:

if (condition) {

if statement

} else {

else statement

}

Конструкция if может быть расширена дополнительным блоком с помощью ключевого слова else.

if (25 < 5) {

console.log('If statement.');

} else {

console.log('Else statement.');

}

Этот блок будет выполнен в том случае, если результат условия if равен false. Если же условие if выполнится, то есть будет равно true, блок else будет пропущен.

Конструкция if…else if…else:

if (condition) {

if statement

} else if (condition) {

else if statement

} else {

else statement

}

Блок else if используют при необходимости проверить нескольких условий.

if (25 < 5) {

console.log('If statement.');

} else if (13 < 23) {

console.log('Else if statement.');

} else {

console.log('Else statement.');

}

При использовании такой конструкции, JavaScript будет выполнять, по очереди, встречающиеся в круглых скобках выражения до тех пор, пока результат не будет равен true. После первого же встретившегося true, остальные блоки условной конструкции будут пропущены. Если же ни одно из условий не будет равно true, выполнится блок else.  
Блок else if может быть использован многократно. Таким образом можно проверять неограниченное количество выражений, однако злоупотреблять таким способом не рекомендуется. Кроме того, блок else не обязательный, условные конструкции if могут строиться и без него:

if (25 < 5) {

console.log('If statement.');

} else if (13 < 1) {

console.log('First else if statement.');

} else if (13 >= 12) {

console.log('Second else if statement.');

}

Следует запомнить, что в результате вычисления выражения может происходить приведение типов, которое будет следовать следующим правилам:  
— число 0, пустая строка «», null, undefined и NaN являются false;  
— остальные значения — true.  
Более детально преобразование типов было рассмотрено в предыдущей лекции.

let test = null;

if (test) {

// will not run

}

let test = "info";

if (test) {

// will run

}

let test = 0;

if (test) {

// will not run

}

Конструкция switch … case

Конструкция switch … case используется для выполнения одного из многих блоков кода на основе сравнения результата выражения с несколькими существующими значениями.

Синтаксис конструкции switch … case:

switch (expression) {

case value1:

// run if result of expression equals value1

break;

case value2:

// run if result of expression equals value2

break;

default:

// run if result not equals none of the values before

}

expression – выражение, которое вычисляется и используется для сравнения.  
case value – ключевое слово (case) указывающее на значение (value), которое необходимо сравнить с результатом выражения (expression).  
break – ключевое слово, которое прекращает дальнейшую проверку на соответствие после первого совпадения.  
default (необязательный параметр) – если подходящего значения не оказалось, выполняется код после ключевого слова default.

let name = "John";

switch (name) {

case "David":

console.log("This is David");

break;

case "John":

console.log("This is John");

break;

default:

console.log("Unknown name");

}

Ключевое слово switch указывает на начало условной конструкции. Сначала выполняется выражение в круглых скобках (expression). Результат выражения, по очереди, сравнивается со значением (value) после ключевого слова case. Как только совпадение найдено, то есть результат сравнения равен true, выполнится соответствующий блок кода. Ключевое слово break прерывает дальнейшую проверку на соответствие после первого же результата true. Если же соответствие не найдено, выполнится код после ключевого слова default.

Если для разных значений (value) необходимо выполнить одинаковый код, можно сгруппировать несколько case:

switch (new Date().getDay()) {

case 0:

case 6:

console.log("It’s weekend");

break;

default:

console.log("It’s weekday");

}

В случае отсутствия ключевого слова break, после найденного соответствия, выход из конструкции switch не произойдет, а это означает, что весь остальной код будет выполнен так же.

let name = "David";

switch (name) {

case "David":

console.log("This is David");

case "John":

console.log("This is John");

default:

console.log("Unknown name");

}

В примере выше, несмотря на то, что соответствие найдено при первом же сравнении, остальные блоки кода будут выполнены также.

Тернарный оператор (? : )

Синтаксис тернарного оператора:

condition ? firstExpression : secondExpression;

condition – условие с результатом true или false.  
firstExpression, secondExpression – выражение, которое выполнится в зависимости от результата условия (condition).

let hours = new Date().getHours();

let greatings = "Good" + ((hours > 17) ? " evening." : " day.");

console.log(greatings);

Если результат выполнения условия будет равен true, то выполнится выражения после оператора ?, если же результат false – выражение после оператора :.  
Выражениями могут быть, например, как присваивание значения переменной (как в примере выше), так и запуск функции:

let value = "old";

(value == "new") ? runNew() : runPrevious();

Тернарный оператор является сокращенным вариантом конструкции if … else.  
Условная конструкция if … else:

let isLoaded;

const data = {statusNumber: 200}if (data.statusNumber > 100) {

isLoaded = "loaded";

} else {

isLoaded = "not loaded";

}

console.log(isLoaded);

Тернарный оператор:

const data = {statusNumber: 200};

let isLoaded = (data.statusNumber > 100) ? "loaded": "not loaded";

console.log(isLoaded);