### ===================== ТЕОРІЯ ============================

### Логічний тип

Логічний тип даних Boolean має всього два значення:

* true («істина», «вірно» або ж «так»);
* false («неправда», «невірно» або ж «ні»).

Отримується в результаті обчислення логічних виразів (виконання операторів порівняння і ін.) та в умовах операторів if , while.

## **Перетворення типів**

Узагальнимо інформацію про систему перетворень типів у JavaScript.

***Перетворення у рядок (String)***

Рядкове перетворення відбувається, коли вимагається подання чого-небудь у вигляді рядка. Наприклад, його генерує функція alert. На практиці для явного перетворення часто застосовується оператор "+", у якого один з аргументів рядок. У цьому випадку він приводить до рядка і інший аргумент. Можна також здійснити перетворення явним викликом String(val).

***Перетворення у число (Number)***

Чисельне перетворення відбувається в математичних функціях і виразах, а також при нестрогому порівнянні даних різних типів. Для перетворення до числа в явному вигляді можна викликати Number(val) , або, покласти перед виразом оператор "+".

|  |  |
| --- | --- |
| **Значення** | **Перетвориться в ...** |
| undefined | NaN |
| null | 0 |
| true | 1 |
| false | 0 |
| Рядок | Пробіли по краях обрізаються. Далі один з випадків:  1) якщо залишається порожній рядок, то 0;  2) з непорожього рядка зчитується число;  3) якщо у рядку не число, то результат NaN . |

***Перетворення до логічного типу даних (Boolean)***

Перетворення до логічного типу (до true/false) відбувається у логічному контексті (там де небхідно перевірити виконання умови), такому як if(obj) , while(obj) і при застосуванні логічних операторів.

|  |  |
| --- | --- |
| **Значення** | **Перетвориться в ...** |
| undefined , null | false |
| Числа | Всі true , крім 0 , NaN - false . |
| Рядки | Всі true , крім порожнього рядка "" - false |
| Об'єкти | Завжди true |

***Логічні вирази***

*Логічний вираз*  -- це вираз в результати одержуємо одне із логічних значень (true/false).

Логічні вирази можуть містити:

* логічні константи (true/false);
* змінні логічного типу;
* оператори порівняння;

У JavScript використовуються такі оператори порівняння

|  |  |
| --- | --- |
| Більше | a > b |
| Менше | a < b . |
| Більше або дорівнює | a >= b |
| Менше або дорівнює | a <= b |
| Рівне  (a=b це присвоювання!) | a == b |
| Не рівно, ≠ | != |
| Тотожно рівно (співпадає тип і значення) | a === b |
| Не тотожно рівно (не співпадає тип, або не співпадають значення) | A !== b |

**При порівнянні за допомогою знаку “==” значення перетворюються до числа (до типу Number)**, крім випадку коли обидва значення є рядками. Порівняння рядків здійснюється у лексикографічному порядку (аналогічно, як ми упорядковуємо слова у алфавітному порядку).

Звичайна рівність (записана за допомогою знаку «==» ) не може відрізнити 0 від false. Оскільки і 0, і false при перетворенні у число дають 0 (а саме це відбувається при порівнянні за знаком «==»). Для перевірки рівності без перетворення типів використовуються оператори тотожної рівності === і !== . Вони порівнюють без приведення типів. Якщо тип різний, то такі значення завжди нерівні.

Для запису більш складних логічних виразів (умов) можна використати логічні операції:

&& -- логічне «і»;

|| -- логічне «або»;

! – логічне заперечення («не»).

Вони називаються «логічними», але в JavaScript можуть застосовуватися до значень будь-якого типу і повертають також значення будь-якого типу.

Розглянемо таблицю значень з результатами застосування даних операторів

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | y | x&&y  (логічне «і») | x || y  (логічне «або») | !x  (заперечення) |
| false | false | false | false | true |
| false | true | false | true | true |
| true | false | false | true | false |
| true | true | true | true | false |

**КОРОТКИЙ ЦИКЛ ОБЧИСЛЕНЬ**

Логічне АБО в класичному програмуванні працює таким чином: «якщо хоча б один з аргументів true, то повертає true, інакше - false».

JavaScript обчислює кілька АБО зліва направо. При цьому, щоб економити ресурси, використовується так званий «короткий цикл обчислення».

Припустимо, обчислюються кілька АБО поспіль: a | | b | | c | | .... Якщо перший аргумент - true, то результат свідомо буде true (хоча б одне із значень - true), і інші значення ігноруються.

Це особливо помітно, коли вираз, передане в якості другого аргументу, має сторонній ефект - наприклад, присвоює змінну.

При запуску прикладу нижче привласнення x не відбудеться:

var x;

true || (x = 1); / / просто обчислимо АБО, без if

alert (x); / / undefined, x не присвоєно

А в прикладі нижче перший аргумент - false, так що АБО спробує обчислити другий, запустивши тим самим присвоювання:

var x;

false || (x = 1);

alert (x); / / 1

Оператор АБО обчислює рівно стільки значень, стільки необхідно - до першого true. Він повертає те значення, на якому зупинилися обчислення.

У класичному програмуванні && повертає true, якщо обидва аргументи істинні, а інакше – false.

До І застосуємо той же принцип «короткого циклу обчислень», але трохи по-іншому, ніж до АБО.  
Якщо лівий аргумент - false, оператор І повертає його і закінчує обчислення. Інакше - обчислює і повертає правий аргумент.

Оператор НЕ спочатку приводить аргумент до логічного типу true / false. Потім повертає протилежне значення.

*Використання короткого циклу обчислень для аналізу значення величини і задання значення за замовчуванням*

var b = parseInt(prompt("b=",""));

var c = b || 17; 🡨 Якщо значення змінної b некоректне, то значення змінної с=17

document.write("c="+c);

## **Умовний оператор**

Іноді, залежно від певних умов, потрібно виконати різні дії. Для цього використовується умовний оператор. Він може бути в повній

|  |  |
| --- | --- |
| Загальна форма | Приклад |
| **if** (умова)  оператор, якщо умова виконується  **else**  оператор, якщо умова не виконується | if(a>b)  max=a;  else  max=b; |

або скороченій формі.

|  |  |
| --- | --- |
| Загальна форма | Приклад |
| **if** (умова)  оператори, якщо умова виконується | max=a;  if(b>max)  max=b; |

Якщо при виконанні чи невиконанні деяких умов треба виконати більше ніж один оператор, то потрібно використовувати складений оператор (блок), який записується за допомогою фігурних дужок

{

оператор1;

оператор2;

. . . . .

}

|  |  |
| --- | --- |
| Загальна форма | Приклад |
| if (умова)  {  оператори, якщо умова виконується  }  else  {  оператори, якщо умова не виконується  } | if(a>b)  {  max=a;  min=b;  }  else  {  max=b;  min=a;  } |

або скороченій формі.

|  |  |
| --- | --- |
| Загальна форма | Приклад |
| if (умова)  {  оператори, якщо умова виконується  } | if(x!=0)  {  z=1/x;  alert(z);  } |

*Приклад.*  Скласти скрипт для запиту від користувача віку. Якщо він <18 років то повідомити що голосувати йому ще рано.

var age=parseInt(prompt('Ваш вік','');

if (age<18)

{

alert('Доступ заборонений');

}

else {

alert('Ласкаво просимо');

};

*Приклад.*  Скласти скрипт який за введеним віком і статтю визначає чи є користувач військовозобов’язаним.

if (!confirm('Ви чоловік?'))

{ alert('жінки не служать');

}

else

{

var age=prompt('Ваш вік','');

if (age==null)

{

alert('Ви не ввели вік')

}

else

{

age=parseInt(age);

if ((age>=18)&&(age<25))

{

alert('служити')} else{alert('не служити')};

}

}

***Тернарний оператор***

Іноді буває ситуація, коли значення виразу повинно дорівнювати одному із двох значень у залежності від виконання чи невиконання деякої умови. Наприклад, потрібно залежно від умови присвоїти змінній певне значення. В цьому випадку можна використати тернарний оператор.

|  |  |
| --- | --- |
| Загальна форма | Змінна = умова ? значення1 (умова викон.) : значення2(умова не викон.);  ----- аналог з умовним оператором ------  if ( умова )  змінна = значення1;  else  змінна = значення2; |
| Приклад | **max= a>b ? a : b;**  ------ аналог з умовним оператором ------  if(a>b)  max=a;  else  max=b; |

*Приклад.*  Скласти скрипт для знаходження більшого з двох чисел.

var a=parseFloat(prompt('Введіть перше число',''));

var b=+prompt('Введіть друге число','');

var max=(a>b)?a:b;

alert('Більше число рівне '+max);

**ОПЕРАТОР ВИБОРУ SWITCH**

Конструкція switch замінює собою відразу кілька if. Це оператор викорситовується тоді, коли у залежності від значення деякої величини (кількість можливих значень є невеликою) потрібно виконати ті чи інші оператори.

Це більш наочний спосіб порівняти вираз (в сенсі оператора «===») відразу з декількома варіантами.

**switch**(селектор) {

**case** значення 1 : оператори   // if (селектор === значення1)

**break;**

**case** значення 2 : оператори // if (селектор === значення2)

**break;**

. . . . . . . . . . . . .

**case** значення n : оператори // if (селектор === значенняn)

**break;**

**default**: //інакше (всі інші значення)

оператори

}

|  |  |
| --- | --- |
| **Загальна форма** | **Приклад.** Вводиться оцінка – цифра, вивести оцінку прописом (селектор вибору цілого типу). |
| **switch (<селектор вибору>)**  **{**  **case** <знач. 1> : <оператор 1>;  **break;**  **case** < знач. 2> : <оператор 2>;  **break;**  **………………………………………….**  **case** < знач. N> : <оператор N>;  **break;**  **default :** <оператор N+1>;  **break;**  **}** | var score = parseInt(prompt("score", ""));  var result;  //-------------- Знаходимо результат за допомогою switch  switch (score)  {  case 2: result="Незадовільно";  break;  case 3: result="Задовільно.";  break;  case 4: result="Добре";  break;  case 5: result="Відмінно";  break;  default: result="Неправильна оцінка.";  break;  }  alert(result); |

Якщо для декількох варіантів необхідно виконати одні і ті ж оператори, то ці оператори вказують тільки для одного з варіантів, а для всіх інших не вказуємо ні необхідних операторів, ні операторів break.

**Приклад**. З клавіатури вводиться оцінка у національній шкалі, необхідно вивести повідомлення про те, чи зараховано студенту залік.

<script>

// З клавіатури ввоидться номер місяця, вивести на екран пору року

var score = parseInt(prompt("score", ""));

var result;

//-------------- Знаходимо результат за допомогою switch

switch (score)

{

case 1:

case 2: result="Незараховано";

break;

case 3:

case 4:

case 5: result="Зараховано";

break;

default: result="Неправильна оцінка.";

break;

}

//-----------------------------

alert(result);

</script>

Слід зазначити, що на відміну від інших С-подібних мов прорамування в якості можливих значень селектора вибору можуть бути величини не тільки порядкового, а і інших типів. Більше того, в якості можливих значень можуть бути використано результат виконання деякого оператора.

**Приклад.**

var b= 1;

var a = 2;

switch (a) {

case 1: document.write(a);

break;

case 3.5: document.write(a); 🡨 Значення є дійсним числом 3.5

break;

case b+1: document.write(a); 🡨 Значення обраховується «b+1»

break;

}

**//==========================ПРИКЛАДИ===========================**

1.1. Найбільше серед двох різних замінити на 0.

1.1.1. Найбільше серед двох замінити на 0. Якщо однакові, то замінити обидва на 1.

1.2. З клавіатури вводиться ціна товару і кількість грошей. Якщо грошей не вистачає то відмовляємо у покупці, інакше, якщо ще залишаються гроші, то пропонуємо купити лотерею за 4 грн.

1.3. Формування накладної. Вводиться назва товару, кількість, ціна і рік. Значення року вводиться користувачем або ж дорівнює 2017 (у випадку, якщо користувач не ввів значення). Перевірити коректність введених даних . У випадку некоректності вивести відповідне повідомлення.

<script>

var product = prompt("Product title") || "Product";

var year = parseInt(prompt("Year")) || 2017;

var count = parseInt(prompt("Count"));

if (count)

//document.write(product + " - " + year + " - " + count);

document.write(`${product} - ${ year} - ${ count}`);

else

document.write("Error");

</script>

===========================

Задача 2. З клавіатури вводиться назва категорії водія. Вивести на екран назву транспортного засобу, яким він може керувати

Задача 2. З клавіатури вводяться два числа. Вибравши номер операції обчислити результат.

Задача 3. Модифікувати попереднє завдання ввівши не номер операції, а саму операцію.

1. На роботу компаніє приймає працівників від 32 до 45 років. З клавіатури вводиться вік претендента. З’ясувати, чи може він бути прийнятим на роботу.
2. З клавіатури вводиться кількість балів. Вивести на екран оцінку (задовільно, добре чи відмінно).

3.1. Комп’ютер випадковим чином генерує число. Вгадати, чи наступне число буде більшим чи меншим.

<script> ДОРОБИТИ!!!

//3.1.Комп’ютер випадковим чином генерує число.

//Вгадати, чи наступне число буде більшим чи меншим.

var firstNumber = Math.random();

document.write("First number = " + firstNumber);

var secondNumber = Math.random();

if (confirm("n 2>n 1") && secondNumber > firstNumber)

{

alert("Wizard")

}

else {

alert("Looooooser");

}

document.write("Second number = " + secondNumber);

</script>

3.2.Випадковим чином генерується число від 1 до 5. Спробуйте вгадати число за 2 спроби

1. З клавіатури вводяться імена двох дітей та кількість шоколадок. Вивести не екран ім’я того учня, у якого кількість шоколадок є більшою.
2. З клавіатури вводиться вік людини. Вивести на екран ким він є (дитиною у садочку, школярем, студентом, працівником, пенсіонером).
3. З клавіатури вводиться поточний час (дві змінні цілого типу: кількість годин і кількість хвилин) протягом якогось із перших 3 уроків (початок о 8.30). З’ясувати, який із трьох триває урок.
4. У двох дітей є цукерки трьох типів «Червоний мак», «Ліщина», «Білочка» (кількість кожного виду цукерок у кожної дитини вводиться з клавіатури). Вивести на екран одне із повідомлень:

1)у однієї із дітей кожного типу цукерок більше;

2)у однієї із дітей двох типів цукерок більше;

3)кількість кожного з видів цукерок однакова.

4)інше.

1. Родина їде на відпочинок автомобілем. З клавіатури вводиться наявна кількість літрів палива, витрати палива на 100 кілометрів та відстань, яку необхідно подолати. Вивести на екран чи потрібно, чи не потрібно буде додатково заправляти автомобіль.
2. Користувач може здійснити покупку у одному із двох відділів («побутова техніка» або «інструменти»). При цьому у кожному із відділів йому будуть пропонувати один із перерахованих товарів:

«побутова техніка»:

* телевізор;
* праска;
* планшет;

«інструменти»

* лопата;
* граблі;
* мітла;
* пила.

Вивести на екран назву товару, яку вибрав користувач.

=============================== ПРИКЛАДИ ОПЕРАТОР ВИБОРУ =============

// З клавіатури вводиться рейтинг користувача. 1-Профі, 2-Досвідчений, 3-Знаючий, 4-Початківець.

//Вивести за рейтингом рядкове представлення.

..=========================================================

Приклад.

// З клавіатури вводиться рейтинг користувача. 1-Профі, 2-Досвідчений, 3-Знаючий, 4-Початківець.

//Чи доволено робити коментарі (1, 2 -дозволено, 3,4- не дозволено)

..=========================================================

Приклад.

// З клавіатури ввоидться номер місяця, вивести на екран пору року

Задача 1. З клавіатури вводиться рівень доступу. Вивести на екран текстове представлення.

1.Аднміністратор

2.Менеджер

3.Редактор

4.Користувач

Задача 1.1. Змінити попереднє завдання вивівши «працівник», якщо це перші 3 пункти.

Задача*.* З використанням операторів вибору реалізувати гру «Камінь ножниці папір» (користувач задає свій варіант, комп’ютер генерує випадковим чином),