10. JavaScript Operators

= is used to assign values.

+ is used to add values.

The assignment operator = is used to assign values to JavaScript variables.

The arithmetic operator + is used to add values together.

Example

Assign values to variables and add them together:

```
y=5;
z=2;
x=y+z;
```

The result of *x* will be:

```
7
```

10.1. JavaScript Arithmetic Operators

Arithmetic operators are used to perform arithmetic between variables and/or values.

Given that **y=5**, the table below explains the arithmetic operators:

Operator	Description	Example	Result of x	Result of y
+	Addition	x=y+2	7	5
-	Subtraction	x=y-2	3	5
*	Multiplication	x=y*2	10	5
/	Division	x=y/2	2.5	5
%	Modulus (division remainder)	x=y%2	1	5
++	Increment	x=++y	6	6
		x=y++	5	6
--	Decrement	x=--y	4	4
		x=y--	5	4

10.2. JavaScript Assignment Operators

Assignment operators are used to assign values to JavaScript variables.

Given that **x=10** and **y=5**, the table below explains the assignment operators:

Operator	Example	Same As	Result
=	x=y		x=5
+=	x+=y	x=x+y	x=15
-=	x-=y	x=x-y	x=5
*=	x*=y	x=x*y	x=50
/=	x/=y	x=x/y	x=2
%=	x%=y	x=x%y	x=0

10.3. The + Operator Used on Strings

The + operator can also be used to add string variables or text values together.

Example

To add two or more string variables together, use the + operator.

```
txt1="What a very";
txt2="nice day";
txt3=txt1+txt2;
```

The result of *txt3* will be:

```
What a verynice day
```

To add a space between the two strings, insert a space into one of the strings:

Example

```
txt1="What a very ";
txt2="nice day";
```

```
txt3=txt1+txt2;
```

The result of *txt3* will be:

```
What a very nice day
```

or insert a space into the expression:

Example

```
txt1="What a very";
txt2="nice day";
txt3=txt1+" "+txt2;
```

The result of *txt3* will be:

```
What a very nice day
```

10.4. Adding Strings and Numbers

Adding two numbers, will return the sum, but adding a number and a string will return a string:

Example

```
x=5+5;
y="5"+5;
z="Hello"+5;
```

The result of *x*,*y*, and *z* will be:

```
10
```

```
55
Hello5
```

The rule is: **If you add a number and a string, the result will be a string!**

10.5. Comparison Operators

Comparison and Logical operators are used to test for *true* or *false*.

Comparison operators are used in logical statements to determine equality or difference between variables or values.

Given that **x=5**, the table below explains the comparison operators:

Operator	Description	Comparing	Returns
==	is equal to	x==8	*false*
		x==5	*true*
===	is exactly equal to (value and type)	x==="5"	*false*
		x===5	*true*
!=	is not equal	x!=8	*true*
!==	is not equal (neither value nor type)	x!=="5"	*true*
		x!==5	*false*
>	is greater than	x>8	*false*
<	is less than	x<8	*true*
>=	is greater than or equal to	x>=8	*false*
<=	is less than or equal to	x<=8	*true*

10.6. How Can it be Used

Comparison operators can be used in conditional statements to compare values and take action depending on the result:

```
if (age<18) x="Too young";
```

10.7. Logical Operators

Logical operators are used to determine the logic between variables or values.

Given that **x=6 and y=3**, the table below explains the logical operators:

Operator	Description	Example
&&	and	(x < 10 && y > 1) is true
||	or	(x==5 || y==5) is false
!	not	!(x==y) is true

10.8. Conditional Operator

JavaScript also contains a conditional operator that assigns a value to a variable based on some condition.

Syntax

```
variablename=(condition)?value1:value2
```

Example

If the variable *age* is a value below 18, the value of the variable *voteable* will be "Too young, otherwise the value of *voteable* will be "Old enough":

```
voteable=(age<18)?"Too young":"Old enough";
```