**1 Built-in functions**

JavaScript ima 5 “built in” funkcija: eval, parseInt, parseFloat, escape, i unescape.

<https://jsconsole.com/>

**1.1 eval**

Evaluira string (nisku) i vraća vrednost.

eval(*Expression*)

*Expression*se evaluira i vraća se rezultat.

**Primeri**

Oba primera imaju 42.

var x = 2

var y = 39

var z = "42"

println(eval("x + y + 1"))

println(eval(z))

**1.2 parseInt**

Parsira string argument i vraća broj (Integer) podrazumevanog brojčajnog sistema ili sistema sa osnovom koja je navedena kao drugi argument.

parseInt(*string*)

parseInt(*string*, *radix*)

*string* is a string that represents the value you want to parse.  
*radix* is an integer that represents the radix of the return value.

The **parseInt** function parses its first argument, a string, and attempts to return an integer of the specified radix (base). For example, a radix of ten indicates to convert to a decimal number, eight octal, sixteen hexadecimal, and so on. For radixes above ten, the letters of the alphabet indicate numerals greater than ninr. For example, for hexadecimal numbers (base sixteen), A through F are used. If a radixes above 36 is specified, **parseInt** returns "NaN."

If **parseInt** encounters a character that is not a numeral in the specified radix, it ignores it and all succeeding characters and returns the integer value parsed up to that point. **parseInt** truncates numbers to integer values.

If the radix is not specified or is specified as zero, JavaScript assumes the following:

* If the input *string* begins with "0x," the radix is sixteen (hexadecimal).
* If the input *string* begins with "0," the radix is eight (octal).
* If the input *string* begins with any other value, the radix is ten (decimal).

If the first character cannot be converted to a number, **parseFloat** returns "NaN".

Primeri (svi vraćaju 15):

parseInt("15")

parseInt("15", 10)

parseInt("F", 16)

parseInt("17", 8)

parseInt("FXX123", 16)

parseInt("1111", 2)

parseInt("15\*3", 10)

parseInt("15\*3blablabla", 10)

Primeri koji vraćaju "NaN" ili nulu:

parseInt("Hello", 8)

parseInt("0x7", 10)

parseInt("FFF", 10)

Primeri – svi vraćaju 17 pošto string počinje sa "0x."

parseInt("0x11", 16)

parseInt("0x11", 0)

parseInt("0x11")

**1.3 parseFloat**

Parsira string argument i vraća broj u pokretnom zarezu (Float).

parseFloat(*string*)

*string* is a String object or literal.

**parseFloat** parses its argument, a string, and returns a floating point number. If it encounters a character other than a sign ( + or -), numeral (0-9), a decimal point, or an exponent, then it returns the value up to that point and ignores that character and all succeeding characters.

If the first character cannot be converted to a number, **parseFloat** returns "NaN".

You can call the **isNaN** function to determine if the result of **parseFloat** is "NaN." If "NaN" is passed on to arithmetic operations, the operation results will also be "NaN."

Primeri koji svi vraćaju 3.14:

parseFloat("3.14")

parseFloat("314e-2")

parseFloat("0.0314E+2")

var x = "3.14"

parseFloat(x)

Primer koji vraća "NaN":

parseFloat("FF2")

## Anonimne funkcije

**Anonimne funkcije** Ukoliko nije potrebno da imenujemo funkciju, već samo da

se izvrši jedanput, možemo koristiti anonimne funkcije.

Kada definišemo funkciju putem dodeljivanja konstantnoj funkciji, izraz sa desne strane jednakosti čini jednu anonimnu

## funkciju, na primer:

function(x) {

return x\*x;

}