

ES5 / p.3

Whyte @ Webdev Club

Numbers

typeof returns *'number'*
Number constructor

double precision floating point / IEEE-754 / 64 bit
double precision doesn't mean *perfect precision*
relative error is $\text{Math.pow}(2, -52) \approx 2.22\text{e-}16$
for accurate calculations, use *integers**

*integer = number that can be written without a fractional component;
example: *2*, *-100*, *2048* are integers, but *3.14*, *-7.18* are not

0-prefix is
deprecated
and doesn't
work with
'use strict'

0o and *0b*
are **es6**-only
prefixes for *octal*
and *binary* number
format

scientific *1e3* or *1e-3*
decimal *123* or *123.0* or *123.*
hexadecimal *0x7b*
octal *0173* or *0o173*
binary *0b1111011*

octals are difficult to read, avoid it

Infinity
NaN
no quotes
unary + for casting
parseInt, parseFloat, isNaN, isFinite
Number, Math

Strings

typeof returns *'string'*

String constructor

"a" or 'a' – no difference

`a` – es6 quotes

'a' or '\x61' or '\u0061'

'3' > '29999'

!0

Boolean values

!1

true

false

Boolean constructor

new Boolean(false)

! operator

!! for casting

Let's continue...

Operators

()

()

++X

--X

X++

X--

()

++X

--X

X++

X--

delete

typeof

void

()

==

++X

!=

--X

===

X++

!==

X--

delete

typeof

void

()

==

++X

!=

--X

===

X++

!==

X--

>

delete

<

typeof

<=

void

>=

() == +

++X !=

--X ===

X++ !==

X-- >

delete <

typeof <=

void >=

()

++X

--X

X++

X--

delete

typeof

void

==

!=

===

!==

>

<

<=

>=

+

-

*

/

%

()

++X

--X

X++

X--

delete

typeof

void

==

!=

===

!==

>

<

<=

>=

+

-

*

/

%

~

!

()

++X

--X

X++

X--

delete

typeof

void

==

!=

===

!==

>

<

<=

>=

+

-

*

/

%

~

!

<<

>>

>>>

()

++X

--X

X++

X--

delete

typeof

void

==

!=

===

!==

>

<

<=

>=

+

-

*

/

%

~

!

<<

>>

>>>

&

|

^

()	==	+	<<
++X	!=	-	>>
--X	===	*	>>>
X++	!==	/	&
X--	>	%	
delete	<	~	^
typeof	<=	!	&&
void	>=		

()

++X

--X

X++

X--

delete

typeof

void

==

!=

===

!==

>

<

<=

>=

+

-

*

/

%

~

!

<<

>>

>>>

&

|

^

&&

||

a?b:c

()	==	+	<<	
++X	!=	-	>>	
--X	===	*	>>>	
X++	!==	/	&	
X--	>	%		
delete	<	~	^	a?b:c
typeof	<=	!	&&	=
void	>=			

()	==	+	<<	
++X	!=	-	>>	
--X	===	*	>>>	
X++	!==	/	&	
X--	>	%		
delete	<	~	^	a?b:c
typeof	<=	!	&&	=
void	>=			,

// end