



→ p.2

‘use strict’

striktní režim → správný režim

Datové typy

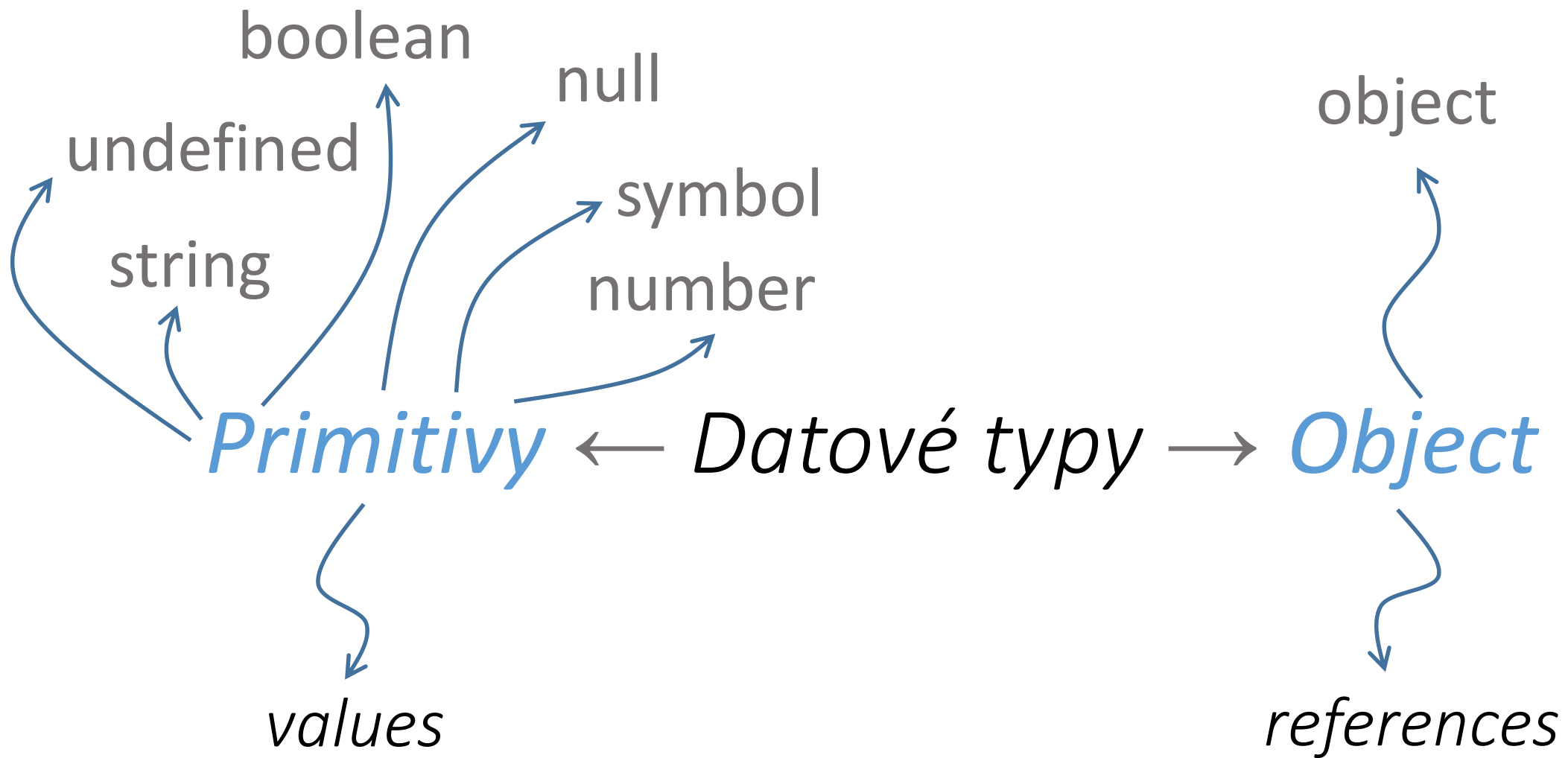


Primitivy ← *Datové typy* → *Object*

Primitivy ← *Datové typy* → *Object*

values *references*

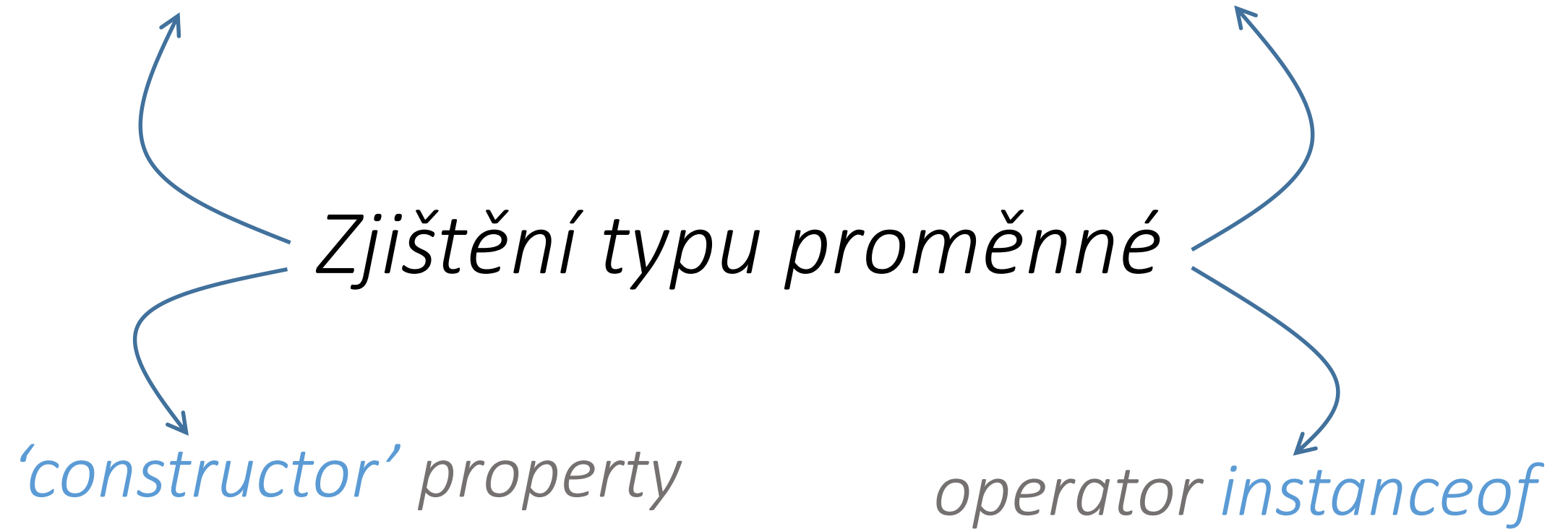
The diagram illustrates the classification of data types in JavaScript. It features three main terms in a horizontal line: 'Primitivy' (in blue italicized font), 'Datové typy' (in black italicized font), and 'Object' (in blue italicized font). 'Primitivy' and 'Object' are connected to 'Datové typy' by grey arrows pointing towards the center. Below 'Primitivy', a blue curved arrow points down to the word 'values'. Similarly, below 'Object', a blue curved arrow points down to the word 'references'.



Zjištění typu proměnné

operator *typeof*

[[class]] property



typeof *x*

vrací string

string → 'string'

number → 'number'

boolean → 'boolean'

symbol → 'symbol'

object → 'object'

array → 'object'

function → 'function'

null → 'object'

Object.prototype.toString.call(n)

má přístup k [[class]]



Object.prototype.toString.call(n)

*vrací jméno konstruktoru jako
“[object Constructor]”*

'constructor' property

nebezpečný způsob

Operator instanceof

nepřesný způsob

Casting

převod mezi datovými typy

s využitím operátorů

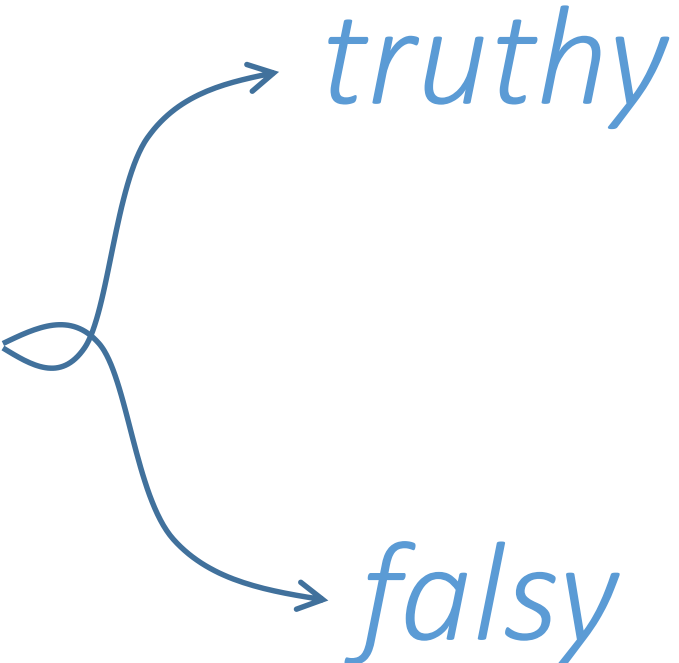
s využitím konstruktorů



Autoboxing

možnost přistupovat na prototypové vlastnosti primitiv

*Všetchny
hodnoty*

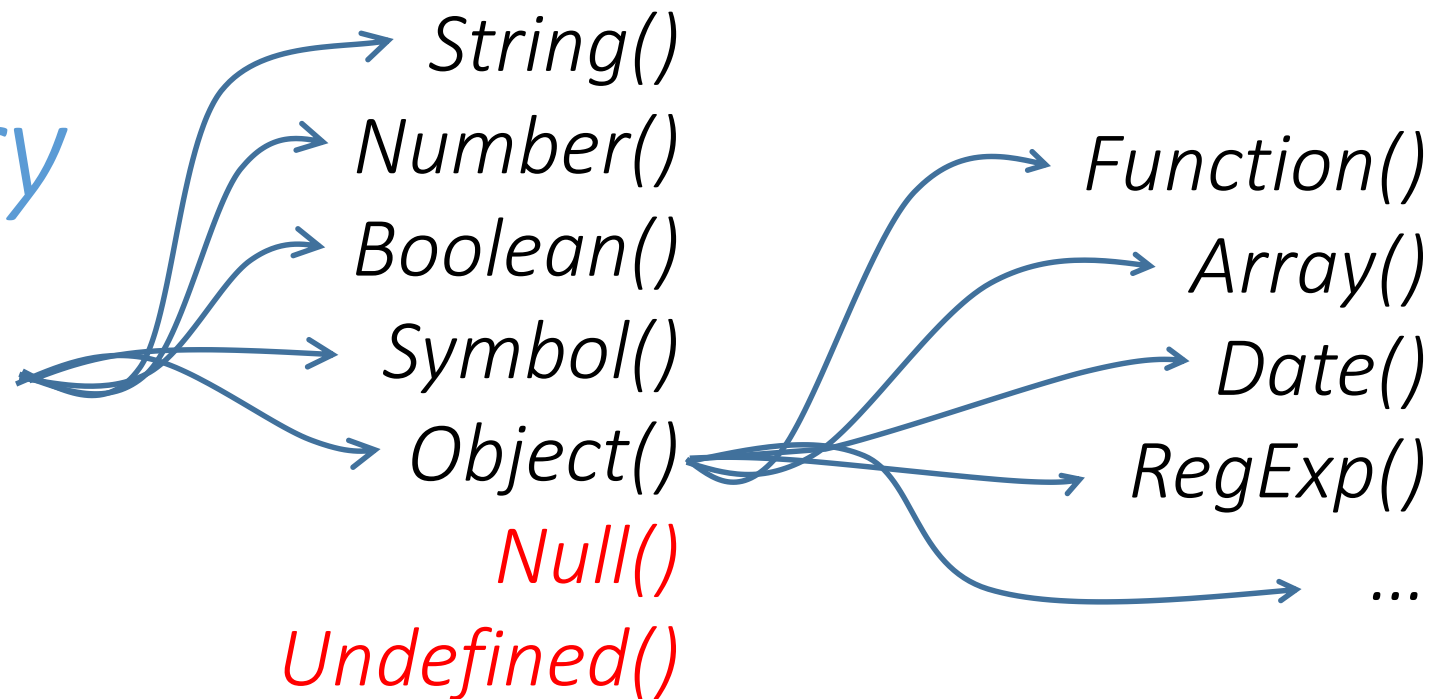


truthy

falsy

Všechny hodnoty { *truthy* = všechny kromě falsy
falsy = null, 0, undefined, NaN, ""

Konstruktory datových typů



Constructor() vs *new Constructor()*

Constructor.prototype

String.prototype.slice = function() {...}

Number



p.2 → *Datové typy* → *Number*

0.1 + 0.2

===

...



0.1 + 0.2

===

0.30000000000000004

IEEE-754

double precision floating point

123 0b1111011 0173 0o173 0x7b

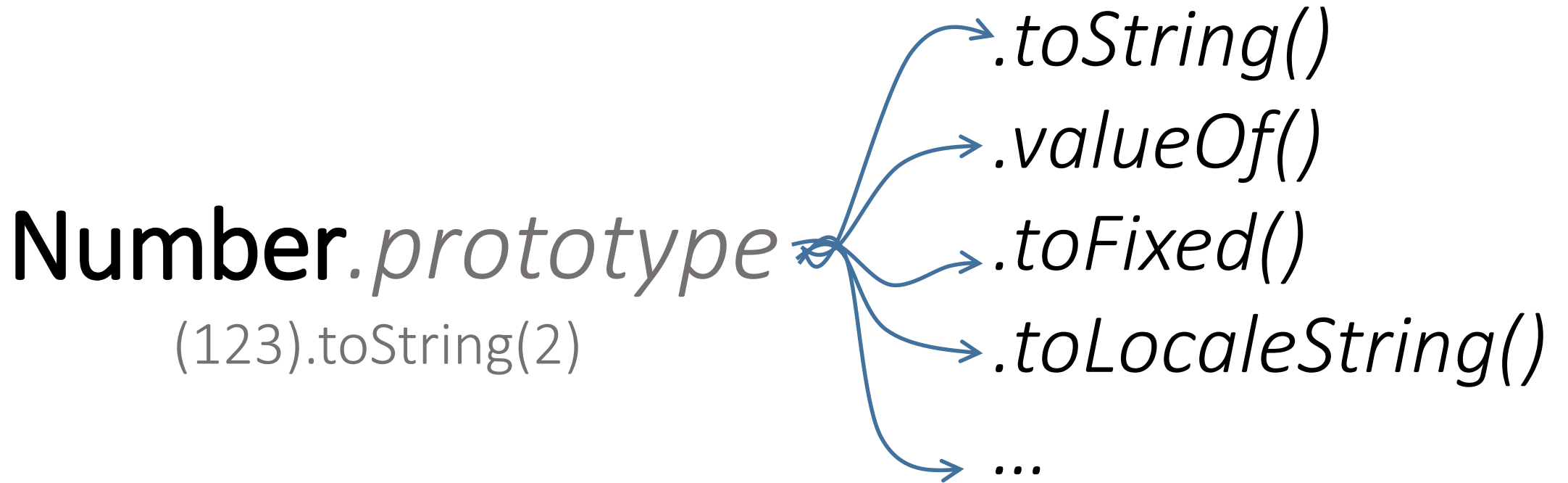
`1e3 === 1000`

`1e-3 === 0.001`

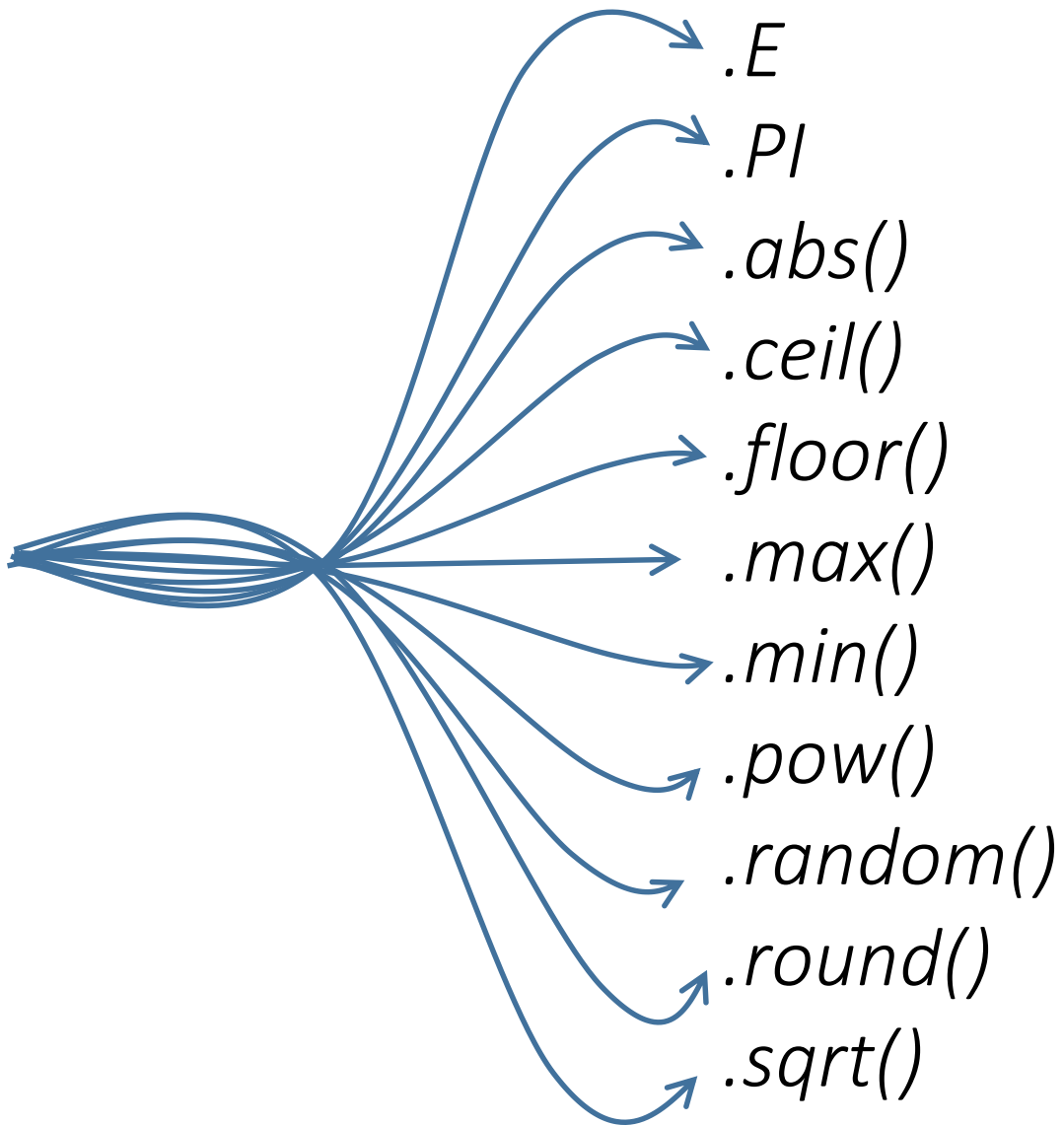
Infinity, NaN

`Number.isFinite()`, `Number.isNaN()`

parseInt(), parseFloat()



Math



String



p.2 → *Datové typy* → *String*

‘abc’

“abc”

`abc`

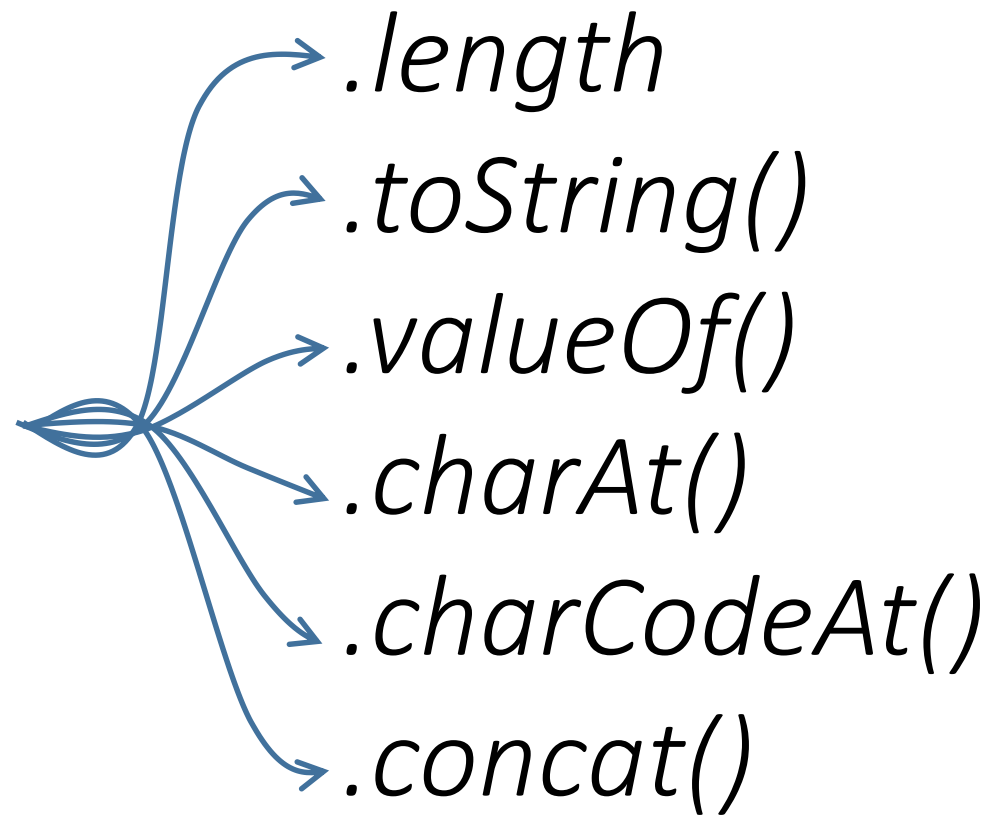
`'\n\t\\'`



`'\x61' === 'a'`
`'\u0061' === 'a'`

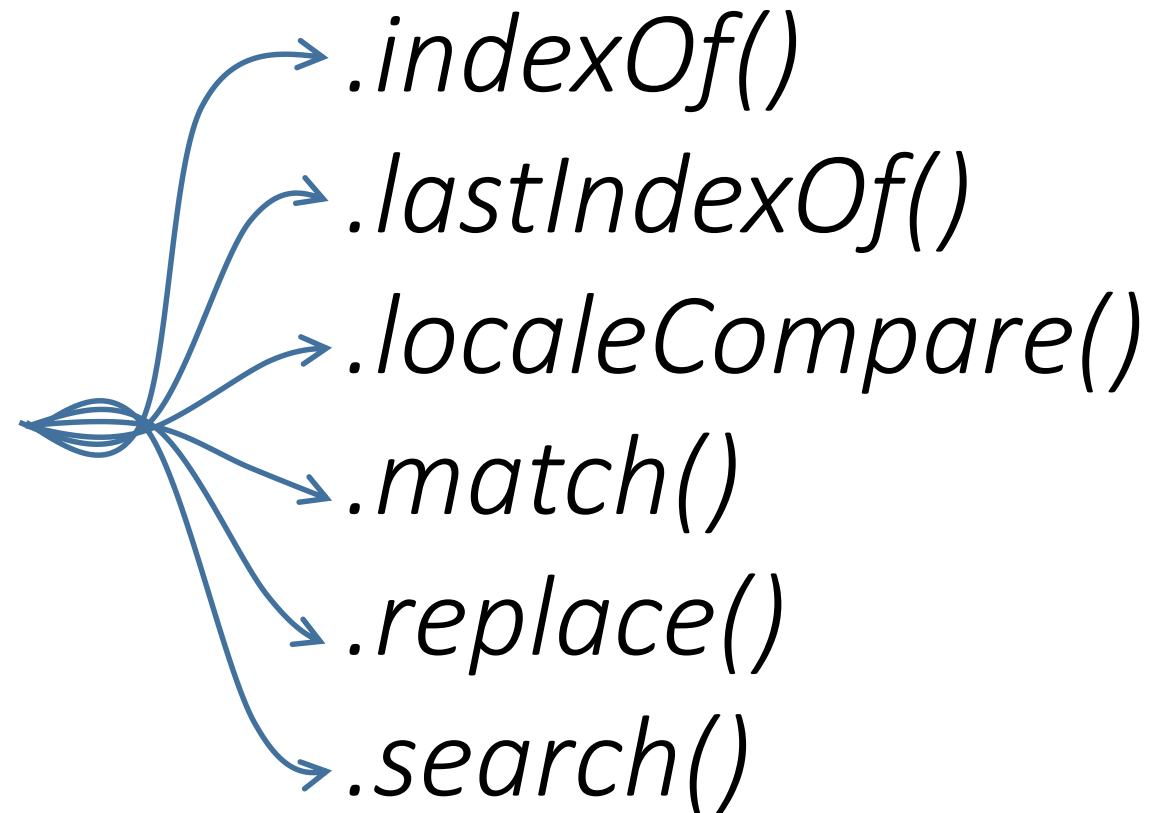
String.prototype

'asd'.charCodeAt(1)



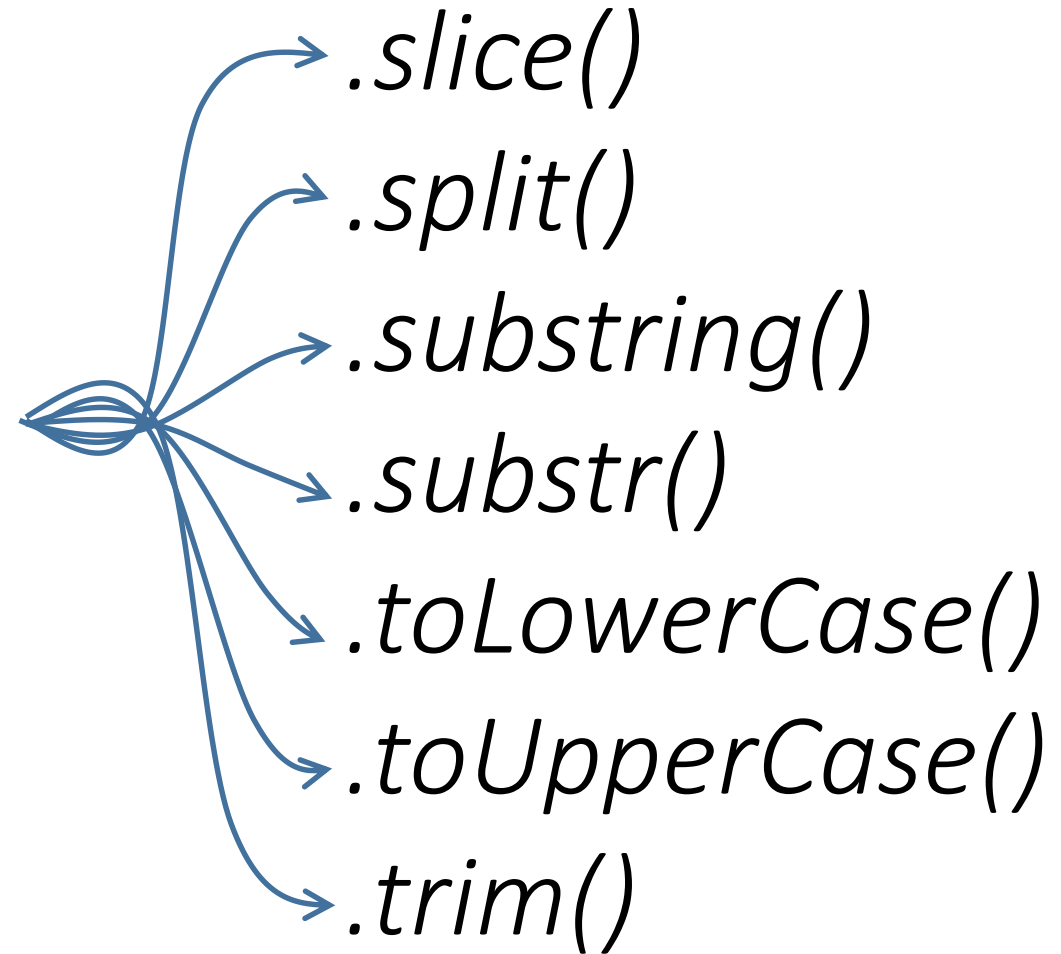
String.prototype

'asd'.charCodeAt(1)



String.prototype

'asd'.charAt(1)



String.*fromCharCode()*

“3” > “299999”

Boolean



p.2 → *Datové typy* → *Boolean*

true, false

new Boolean

Úkoly → bit.ly/2XvFkTp

// end