



→ p.4

# *Function*



p.4 → *Funkce*

*typeof* function(){} === '*function*'

*Pozor, datový typ funkce  
je ale **object**!*



*typeof* function(){} === '**function**'

*funkce má všechny objektové vlastnosti*

*...má i některé navíc, například skryté vlastnosti `[[Call]]` a `[[Scope]]`*

# *Scope*

*funkce je inkapsulační jednotka, vymezuje viditelnost proměnných*

# *Globální vs. Lokální*



*window* – prohlížeč

*global* – Node.js



## *Globální vs. Lokální*



*window* – prohlížeč

*global* – Node.js



## *Globální vs. Lokální*



*LexicalEnvironment*

# *LexicalEnvironment*

*interní objekt (není pro nás přímo přístupný),  
obsahující lokální proměnné + argumenty*

$fn.[[Scope]] = LE$  “rodičovské” funkce

*Hoisting* → *var, function*




p.4 → *Funkce* → *Hoisting*

*Temporal Dead Zone* → *let, const*

# *Function*

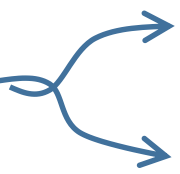


p.4 → *Funkce* → *Druhy funkcí*


*Function*  *Declaration*



p.4 → *Funkce* → *Druhy funkcí*

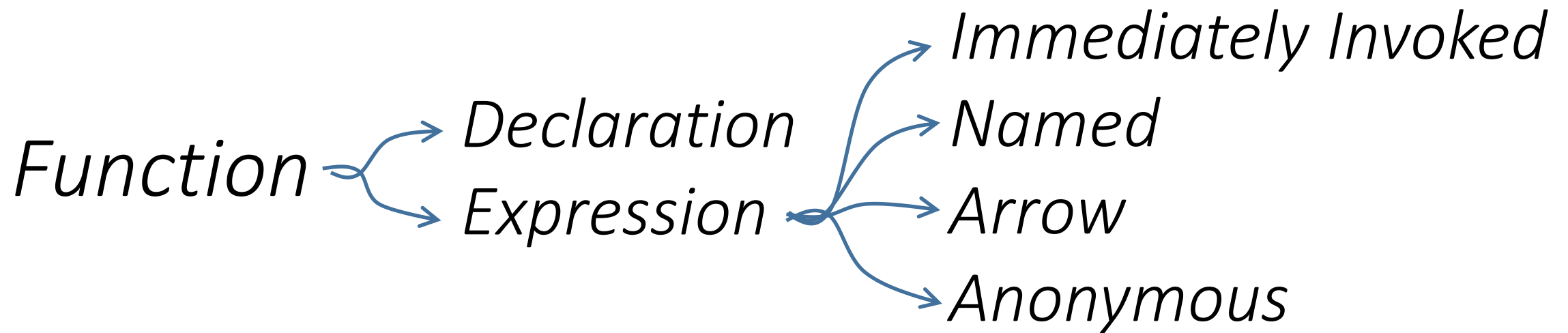
*Function*  *Declaration*  
*Expression*



*Function*  *Declaration*  
*Expression* *Immediately Invoked*







# *Function constructor*



p.4 → *Funkce* → *Function constructor*

# *argumenty*

*Array-like object*



p.4 → *Funkce* → *Argumenty*

# *length*

*počet pojmenovaných argumentů*

**JS**

p.4 → *Funkce* → *Function length*

*name*

*anonymní funkce = funkce, která nemá název*





*const f = function() {}*

*je to pořád anonymní funkce*



```
const f = function() {}
```



*přiřazení neznamená pojmenování*

*pojmenovaná funkce*



```
const f = function f() {}
```

# *Podmíněná definice*

*nemá smysl u function declaration*

# *Rekurze*

*funkce znovu volána dříve, než je dokončeno její předchozí volání*

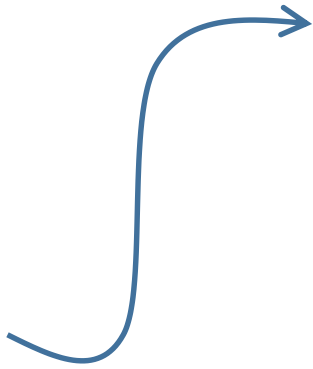
# *Callback*

*funkce jako hodnota, která je volána později*

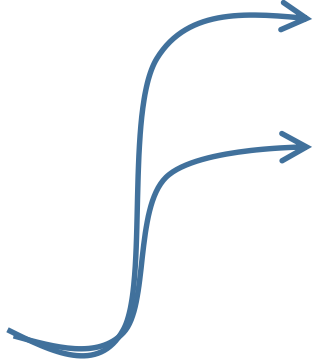
# *Closure*

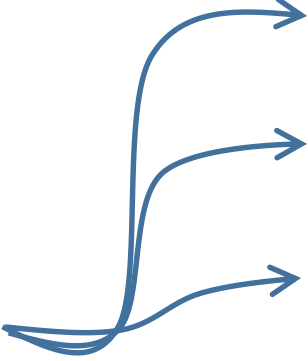


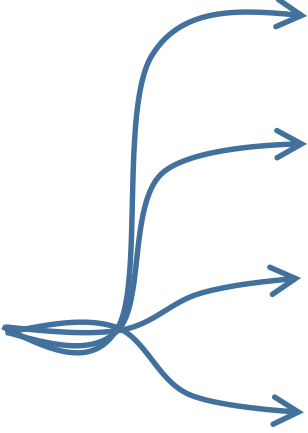
p.4 → *Funkce* → *Closure*

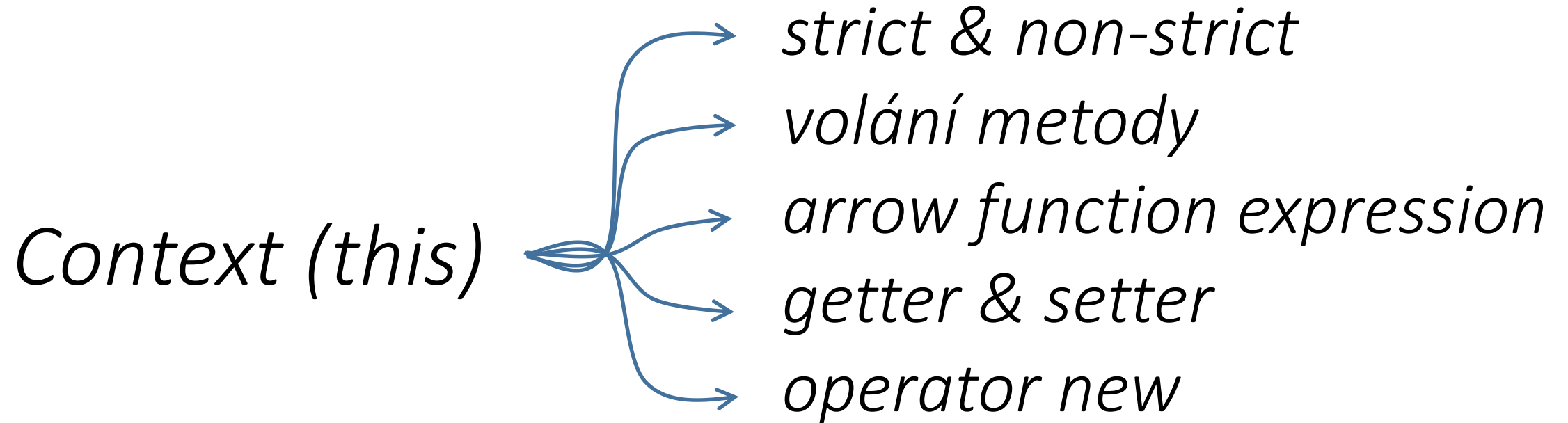
*Context (this)*  *strict & non-strict*

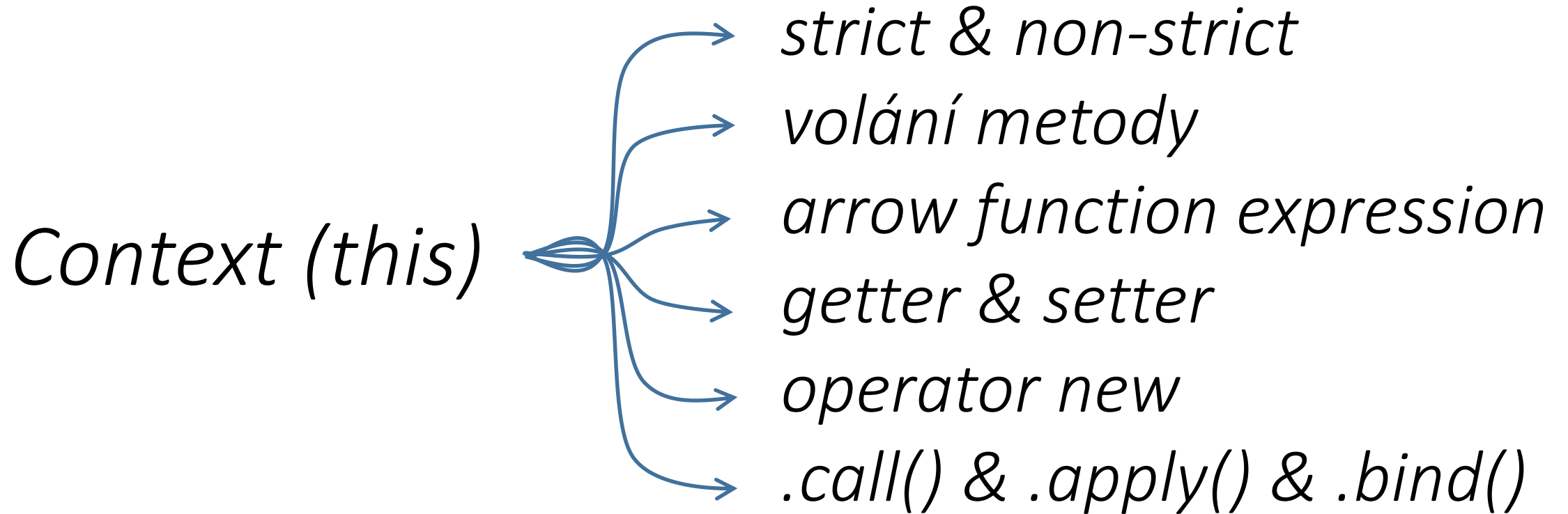


*Context (this)*  *strict & non-strict*  
*volání metody*

*Context (this)*  *strict & non-strict*  
*volání metody*  
*arrow function expression*

*Context (this)*  *strict & non-strict*  
*volání metody*  
*arrow function expression*  
*getter & setter*





*Úkoly* → [bit.ly/2IX20Jh](https://bit.ly/2IX20Jh)

// end