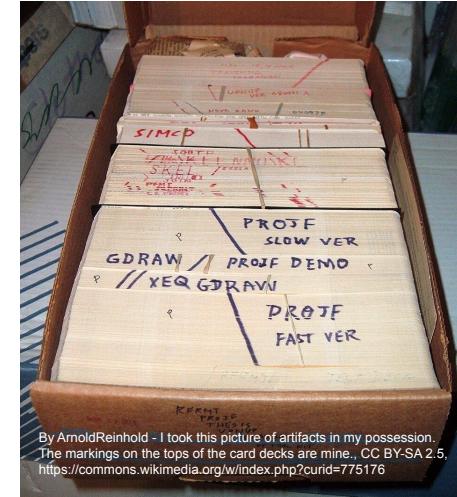


Programování

Erik

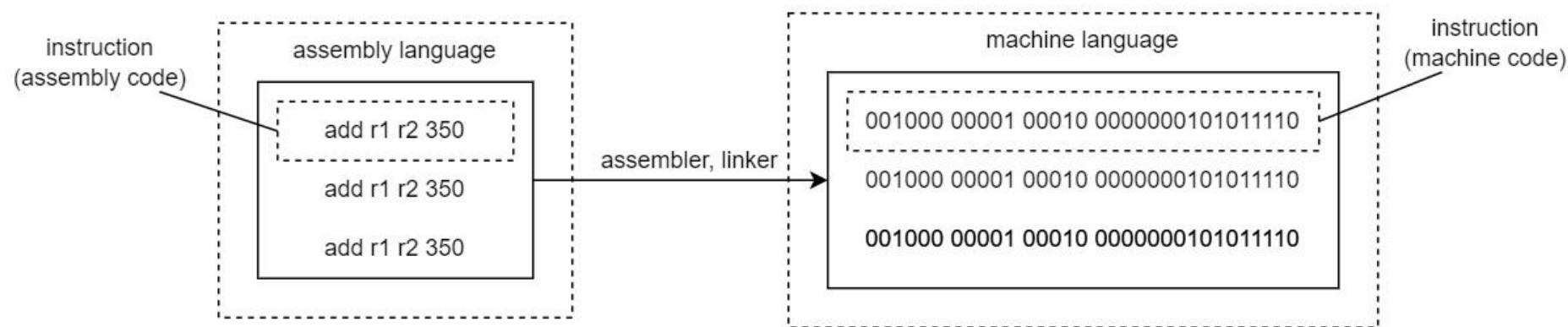
Historie, aneb co, proč a jak?

- Programovat počítač = donutit ho dělat, co chceme
- Von Neumann: Dráty a přepínače → strojový kód
- Počítačový program = sada instrukcí pro počítač, která popisuje automatizovanou realizaci úlohy



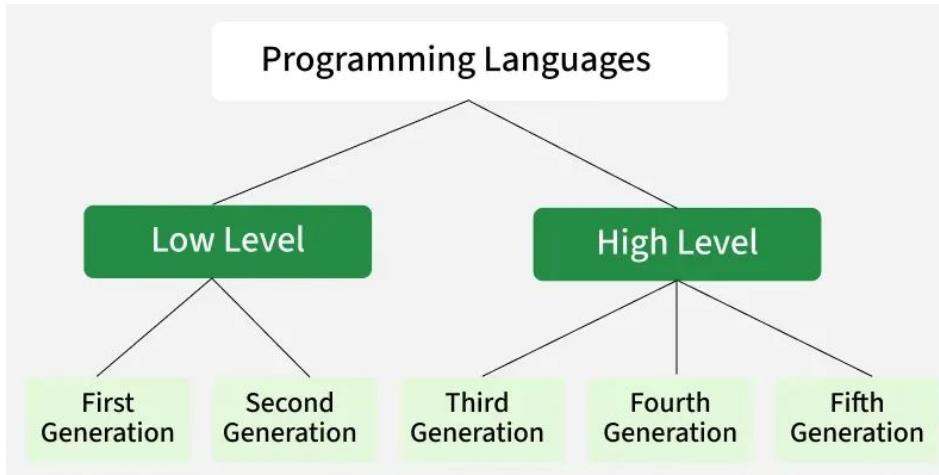
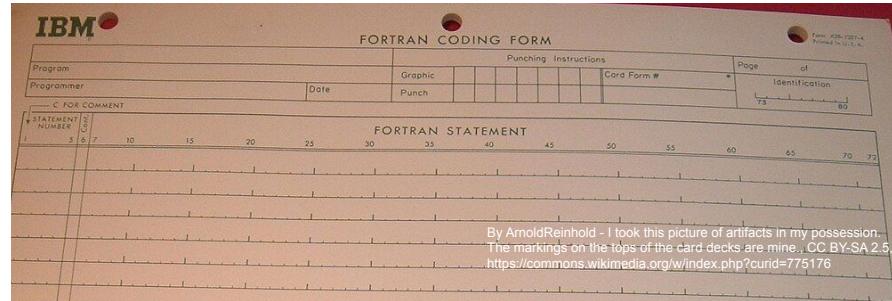
Jiný zápis strojového kódu

- ASSEMBLY = písmenka a čísla 2-9 (označuje prvky symbolickými popiskami)
- kód/instrukce = 1/1
- stroj s jinou instrukční sadou má i jinou ASSEMBLY



Programovací jazyk

- Další úroveň abstrakce!!!
- "Most" mezi strojovým kódem a lidským uvažováním
- Rozdíl mezi strojovým kódem a zdrojovým kódem
- Využití překladače (assembler → compiler)

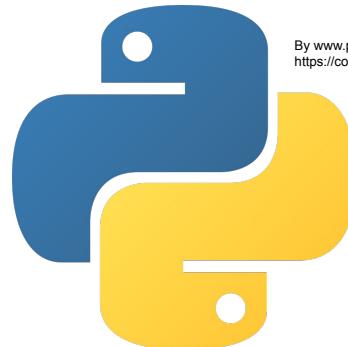
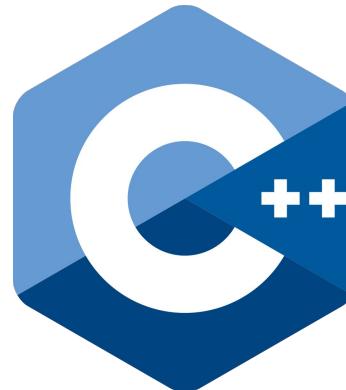


```
HELLO COBOL I1 F 80 TRUNC=72 SIZE=7 LINE=0 COLUMN=1  
00000 * * * TOP OF FILE * * *  
00001 IDENTIFICATION DIVISION.  
00002 PROGRAM-ID. HELLO.  
00003 ENVIRONMENT DIVISION.  
00004 DATA DIVISION.  
00005 PROCEDURE DIVISION.  
00006 DISPLAY "HELLO WORLD." .  
00007 STOP RUN.  
00008 * * * END OF FILE * * *
```

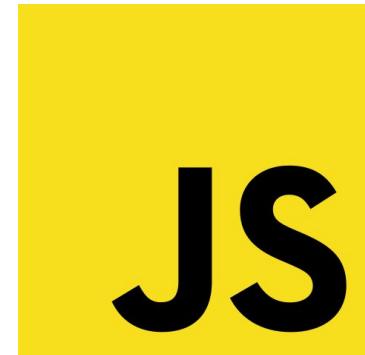
By WEC3390 - Own work, CC BY-SA 4.0,
<https://commons.wikimedia.org/w/index.php?curid=121297364>

Způsoby překladu

- Compiler (překladač)
- Interpreter (interpret)
- Just-in-time komplikace

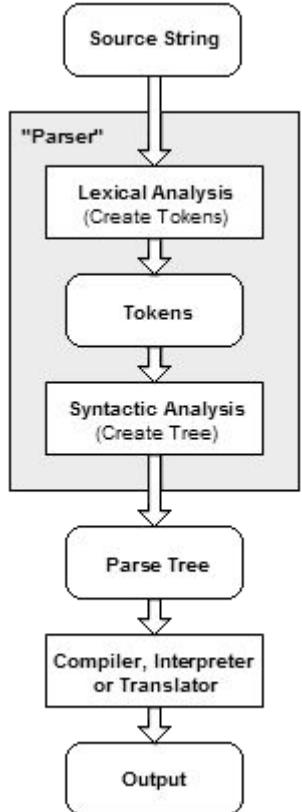


By www.python.org - www.python.org, GPL,
<https://commons.wikimedia.org/w/index.php?curid=34991651>



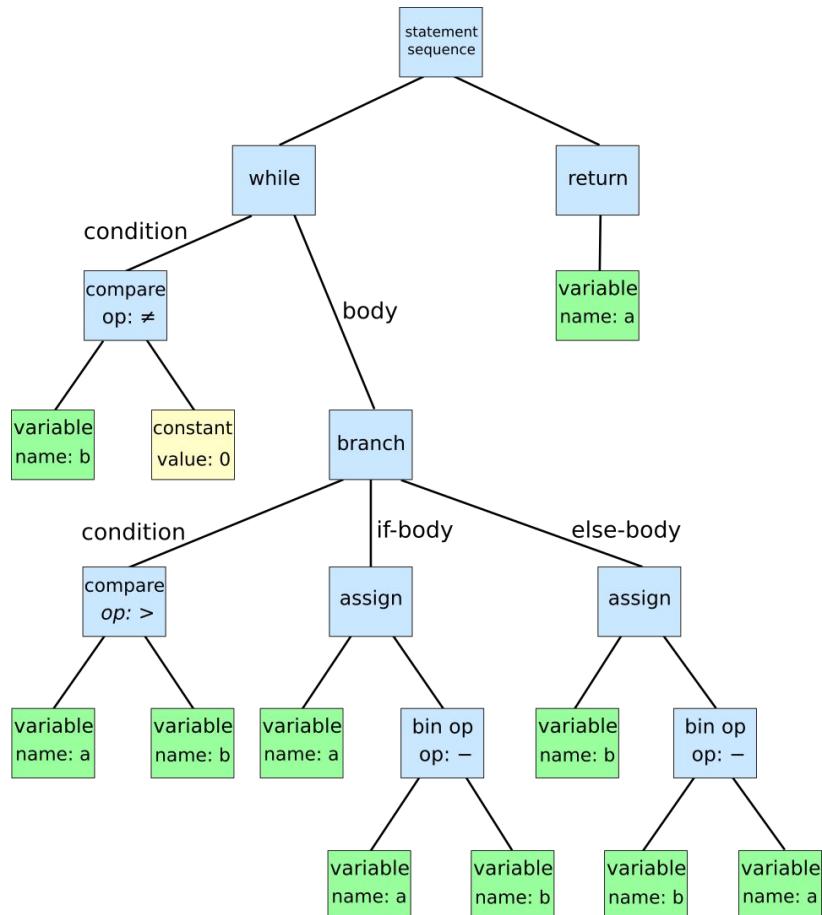
Autor: JavaScript Corp. – The javascript foundation, CC BY-SA 4.0,
<https://commons.wikimedia.org/w/index.php?curid=103235118>

Syntax a sémantika



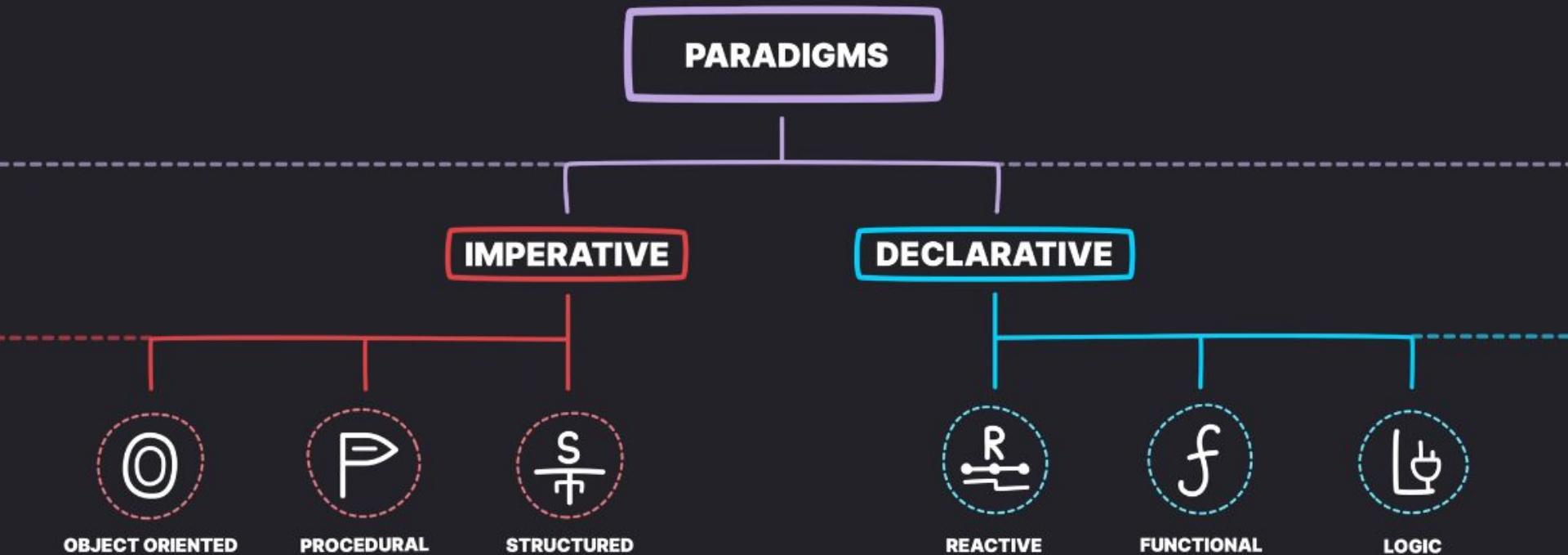
- Kompilátor/interpret vám nemusí vždy rozumět

```
while b != 0:  
    if a > b:  
        a := a - b  
    else:  
        b := b - a  
return a
```



warning: add explicit braces to avoid dangling else

Paradigmata



konec