

Imperative programming

System of rules in programming languages

Tamás Kozsik et al

Eötvös Loránd University

September 20, 2022



Rules of programming languages

- Lexical
- Syntactic
- Semantic

What building blocks does a language have?

- Keywords: while, for, if, else, etc.
- Operators: +, *, ++, ?:, etc.
- Parentheses and separation characters
- Literals: 42, 123.4, 44.44e4, "Hello World!", etc.
- Identifiers
- Notes

Case-(in)sensitive?

Identifier

- Alphanumeric
- Shouldn't start with a number
- Can it contain `_`?

Good

- `factorial`, `i`
- `compitePi`, `open_file`, `worth2see`, `Z00`
- `__main__`

Bad

- `2cents`, `2e3`
- `fifty%`
- `nőnemű`, `Αθήνα` (good in Java, though)

Syntactic rules

- How to build a program?
- How is a branch or loop made?
- How does a sub-program look like?
- etc.

Good

```
if (x != 0)
    y = 1 / x;

int triple(int x) {
    return 3 * x;
}
```

Bad

```
if x != 0
    y = 1 / x

triple(x) {
    return 3 * x;
}
```

Backus-Naur form (Backus normal form) – BNF

```
<statement> ::= <expression-statement>  
              | <while-statement>  
              | <if-statement>  
              | ...
```

```
<while-statement> ::= while (<expression>) <statement>
```

```
<if-statement> ::= if (<expression>) <statement>  
                  <optional-else-part>
```

```
<optional-else-part> ::= "  
                        | else <statement>
```

Semantic rules

- Does it make sense that we built?
- Have I passed correct parameters to operations?
- etc.

Good

```
3 + 3.14    // Bad in Ada
```

Rossz

```
"hello" + 42
```

- How can we express ourselves effectively?
- Conventions
 - Placement of curly braces
 - Naming (e.g. setter/getter)
 - Spelling and language of identifiers
 - Lowercase and uppercase
 - Length of line
- Idioms
- Good and bad practices