
Setanta syntax outline

The syntax of *Setanta* is new, but should feel familiar to most people. It has been designed to be simple and approachable. It takes inspiration from C like languages, but has some new ideas of it's own. The grammar has been designed for unambiguity so no semicolons or similar construct are required.

Setanta programs, like most imperative languages, consist of a sequence of statements. Some important *Setanta* features are outlined below:

- **Variable declarations**

In *Setanta* variables are declared using the `:=` operator, and can be re-assigned using the classic `=` operator. The distinction is to provide a clear lexical difference between variable declaration and reassignment.

Listing A.1: Variables

```
1 x := 0
2 x = x + 1
```

- **Conditionals**

Setanta support the classic conditional execution structure of if, else if, else. This is mostly a direct translation into Irish, as it uses the keyword **má** meaning “if”. However it should be noted that no bracketing is required around the expression.

Listing A.2: Setanta conditionals

```
1 má x == 0
2     scríobh('Tá x cothrom le 0')
3 nó má x == 1
4     scríobh('Tá x cothrom le 1')
5 nó
6     scríobh('Tá x níos mo ná 1')
```

- **Loops**

Setanta supports two main types of loops, “le idir” loops that allow the user to specify start and ends to the loop, and “nuair-a” loops, which are the familiar while loops.

Listing A.3: Setanta loops

```
1 i := 0
2 le i idir (0, 10)
3     i = i + 1
4 x := 0
5 nuair-a x < 10
6     x = x + 1
```

- **Functions**

Functions in *Setanta* are referred to with the term “gníomh” meaning “action”. They can constructed with the **gníomh** keyword, followed by a name and argument list. The **toradh** keyword can be used to return values from the function.

Listing A.4: Setanta function example

```
1 gníomh fibonacci(n) {  
2     má n <= 1  
3         toradh 1  
4     toradh fibonacci(n-1) + fibonacci(n-2)  
5 }
```

- **Classes**

Setanta supports declaring new classes, with methods, and a constructor. Classes can inherit from other classes using the keyword *ó*

Listing A.5: Setanta classe example

```
1 creatlach Person ó Animal {  
2     gníomh nua(name) {  
3         name@seo = name  
4     }  
5     gníomh speak() {  
6         scríobh('Hi, My name is ' + name@seo)  
7     }  
8 }
```

- **Literals**

Setanta supports literals for integers, booleans, null, strings and lists.

Listing A.6: Setanta literals

```
1 a := 500  
2 b := 'Dia duit domhan'  
3 c := [1,2,3,4, fíor ]  
4 d := fíor != breag  
5 c := neamhní
```