

# Namespace: Language

## Headline

The namespace of [software languages](#)

## Description

This namespace is dedicated to [software languages](#), e.g., programming languages such as [Language:Java](#), [Language:Python](#), and [Language:Haskell](#). Software languages are supposed to be described already elsewhere on the web, e.g., on Wikipedia, and thus, 101wiki coverage can be minimalistic and the 101wiki page for a language serves essentially as a [linked open data](#) resource. Metadata for a language should serve the classification of languages on 101wiki.

## Metadata

- [Software language](#)
  - [Section:Headline](#)
  - [Section:Details](#)
  - [Section:Quote](#)
  - [Section:Illustration](#)
  - [Section:Metadata](#)
  - [Property:instanceOf](#)
  - [Language:Haskell](#)
-

# Language: Java

## Headline

An [OO programming language](#)

## Illustration

Let's show "Hello World" for Java.

```
public class HelloWorld {  
  
    public static void main(String[] args) {  
        System.out.println("Hello, World");  
    }  
  
}
```

## Metadata

- [OO programming language](#)
  - [OO programming](#)
  - [https://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))
  - <https://java.com>
  - [Technology:Java platform](#)
  - [Technology:Java SE](#)
-

# Concept: Software language

## Headline

A software language

## Quote

The following quote, which was extracted on 1 March 2011 from the website <http://www.sleconf.org/2011/> of "The International Conference on Software Language Engineering (SLE 2011)", serves as the approximation of a description: *"The term 'software language' comprises all sorts of artificial languages used in software development including general-purpose programming languages, domain-specific languages, modeling and meta-modeling languages, data models, and ontologies. Used in its broadest sense, examples include modeling languages such as UML-based and domain-specific modeling languages, business process modeling languages, and web application modeling languages. The term 'software language' also comprises APIs and collections of design patterns that are implicitly defined languages."*

## Metadata

- <http://www.sleconf.org/2011/>
  - [Vocabulary:Software language engineering](#)
-

# Concept: **Linked open data**

## Headline

A linked network of information that is publicly available

## Quote

*"Linked Open Data (LOD) is Linked Data which is released under an open license, which does not impede its reuse for free."*

â€” Tim Berners-Lee, Linked Data

## Illustration

See <https://www.wikidata.org/wiki/Q251> or <https://www.wikidata.org/wiki/Q251>

Actually any semantic wiki, such as the 101wiki also contains linked open data.

## Metadata

- [https://en.wikipedia.org/wiki/Linked\\_data#Linked\\_open\\_data](https://en.wikipedia.org/wiki/Linked_data#Linked_open_data)
  - <https://www.w3.org/DesignIssues/LinkedData.html>
-

# Language: Haskell

## Headline

The [functional programming language](#) Haskell

## Details

101wiki hosts plenty of Haskell-based contributions. This is evident from corresponding back-links. More selective sets of Haskell-based contributions are organized in themes: [Theme:Haskell data](#), [Theme:Haskell potpourri](#), and [Theme:Haskell genericity](#). Haskell is also the language of choice for a course supported by 101wiki: [Course:Lambdas\\_in\\_Koblenz](#).

## Illustration

The following expression takes the first 42 elements of the infinite list of natural numbers:

```
> take 42 [0..]
[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41]
```

In this example, we leverage Haskell's [lazy evaluation](#).

## Metadata

- <http://www.haskell.org/>
  - [http://en.wikipedia.org/wiki/Haskell\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/Haskell_(programming_language))
  - [Functional programming language](#)
-

# Language: Python

## Headline

A [multi-paradigm programming language](#)

## Metadata

- [OO programming language](#)
  - [Interpreted programming language](#)
  - [Functional programming language](#)
  - [Multi-paradigm programming language](#)
  - [http://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/Python_(programming_language))
  - <http://wiki.python.org/moin/FrontPage?action=show&redirect=StartSeite>
  - [dictionary](#)
-