

# Python

<http://programming.dojo.net.nz/python>

言語道場

```
for i in xrange(1, 101):  
    if i % 15 == 0:  
        print "FizzBuzz"  
    elif i % 3 == 0:  
        print "Fizz"  
    elif i % 5 == 0:  
        print "Buzz"  
    else:  
        print i
```

Imperative

Object Oriented

Dynamic Typing

Cross Platform

Web Development

Network Development

Game Development

Application Development

# Ruby

<http://programming.dojo.net.nz/ruby>

言語道場

```
1.upto(100) do |n|  
  print "Fizz" if a = (n % 3).zero?  
  print "Buzz" if b = (n % 5).zero?  
  print n unless (a || b)  
  print "\n"  
end
```

Imperative

Object Oriented

Dynamic Typing

Cross Platform

Web Development

Network Development

Game Development

Application Development

# C

<http://programming.dojo.net.nz/c>

## 言語道場

```
#include <stdio.h>

int main (void)
{
    int i;
    for (i = 1; i <= 100; i++)
    {
        if (!(i % 15))
            printf ("FizzBuzz\n");
        else if (!(i % 3))
            printf ("Fizz\n");
        else if (!(i % 5))
            printf ("Buzz\n");
        else
            printf ("%d\n", i);
    }
    return 0;
}
```

Imperative

Procedural

Legacy

Statically Typed

High Performance

Operating Systems  
Development

Game Development

Embedded Development

# Java

<http://programming.dojo.net.nz/java>

言語道場

```
public class FizzBuzz {
    public static void main (String[] args) {
        for (int i= 1; i <= 100; i++) {
            if (i % 15 == 0) {
                System.out.println("FizzBuzz");
            } else if (i % 3 == 0) {
                System.out.println("Fizz");
            } else if (i % 5 == 0) {
                System.out.println("Buzz");
            } else {
                System.out.println(i);
            }
        }
    }
}
```

Imperative

Object Oriented

Cross Platform

Statically Typed

Business Oriented

Networked Applications

Embedded Development

Application Development

# Scheme

<http://programming.dojo.net.nz/scheme>

言語道場

```
(do ((i 1 (+ i 1)))  
    ((> i 100))  
    (display  
      (cond ((= 0 (modulo i 15)) "FizzBuzz")  
            ((= 0 (modulo i 3)) "Fizz")  
            ((= 0 (modulo i 5)) "Buzz")  
            (else i))))  
    (newline))
```

Functional

Homoiconic

Cross Platform

Dynamically Typed

Minimalist

Information Processing

Web Development

Academic Research

# Basic

<http://programming.dojo.net.nz/basic>

言語道場

```
FOR A = 1 TO 100
  IF A MOD 15 = 0 THEN
    PRINT "FizzBuzz"
  ELSE IF A MOD 3 = 0 THEN
    PRINT "Fizz"
  ELSE IF A MOD 5 = 0 THEN
    PRINT "Buzz"
  ELSE
    PRINT A
  END IF
NEXT A
```

Unstructured

Procedural

Dynamically Typed

Many Implementations

Simple Semantics

Web Development

Application Development

Game Development