Python

programming.dojo.net.nz/python

```python
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
```

Interpreted

Object Oriented

Dynamic Typing

Cross Platform

Web Development

Network Development

Game Development

Application Development

Ruby

programming.dojo.net.nz/ruby

```ruby
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
```

Interpreted

Object Oriented

Dynamic Typing

Cross Platform

Web Development

Network Development

Game Development

Application Development

C

programming.dojo.net.nz/clang

```c
#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

Low Level

Compiled

Statically Typed

High Performance

Operating Systems Development

Game Development

Embedded Development

Java

programming.dojo.net.nz/java

```java
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);
      }
    }
  }
}
```

Object Oriented

Compiled

Cross Platform

Statically Typed

Business Oriented

Networked Applications

Embedded Development

Application Development

Scheme

programming.dojo.net.nz/scheme

```scheme
(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

Interpreted

Cross Platform

Dynamically Typed

Homoiconic

Information Processing

Web Development

Academic Research

Visual Basic

programming.dojo.net.nz/basic

```
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
```

Imperative

Interpreted

Proprietary

Dynamically Typed