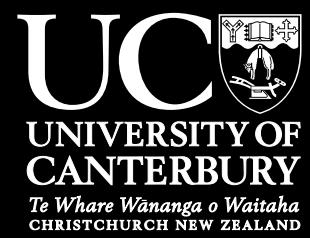


Python

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



言語道場

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

First appeared in

1991

Popular for

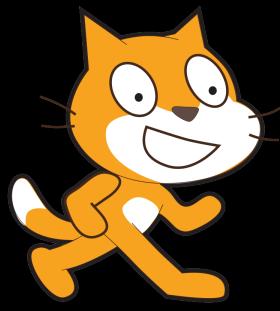
Computer Science
Education,
Internet, Scripting,
Games

Major paradigms

Imperative,
Object Oriented,
Dynamic Typing

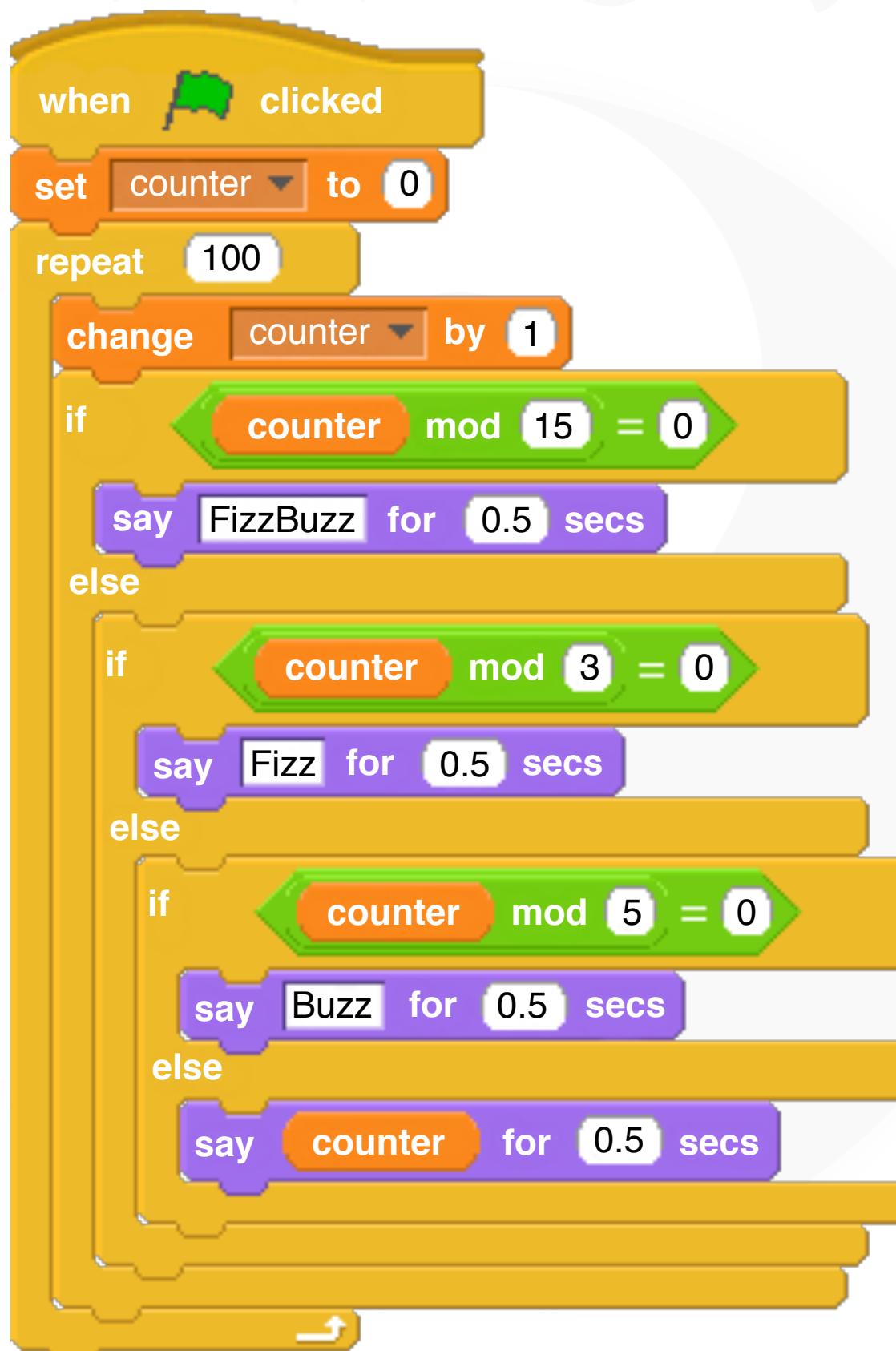
Features

Cross Platform,
Duck Typing,
Indentation Syntax,
Interpreter



SCRATCH

<http://programming dojo net nz scratch>



First appeared in

2007

Popular for

Computer Science Education

Major paradigms

Fixed Function,
Imperative, Visual Programming

Features

Rich Media, Online Collaboration, Animation, Cross Platform



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
```

First appeared in

1995

Popular for

Internet, Scripting

Major paradigms

Imperative,
Object Oriented,
Dynamic Typing,
Functional

Features

Cross Platform,
Duck Typing,
Programmer Happiness

C

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

```
#include <stdio.h>

int main (int argc, char** argv)
{
    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;
}
```

First appeared in

1972

Popular for

Operating Systems,
Compilers, Interpreters,
Embedded Processors,
Games

Major paradigms

Imperative,
Static Typing

Features

High Performance, Low
Level, Pervasive



Java

<http://programming dojo.net.nz/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);
            }
        }
    }
}
```

First appeared in

1995

Popular for

Applications, Mobile Devices, Compilers, Interpreters, Games

Major paradigms

Imperative, Object Oriented, Static Typing

Features

Interoperability, Standardised, Cross Platform

C#

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

```
using System;

namespace FizzBuzz
{
    class Program
    {
        static void Main(string[] args)
        {
            for (int i = 1; i <= 100; i++)
            {
                string output = "";
                if (i % 3 == 0) output += "Fizz";
                if (i % 5 == 0) output += "Buzz";
                if (String.IsNullOrEmpty(output))
                    output = i.ToString();
                Console.WriteLine(output);
            }
        }
    }
}
```

First appeared in

2001

Popular for

Applications, Internet,
Business, Games

Major paradigms

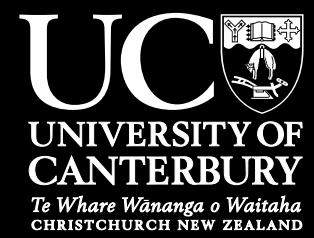
Imperative,
Object Oriented,
Static Typing

Features

Standardised

Scheme

<http://programming dojo.net.nz/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)
)
```

First appeared in

1975

Popular for

Computer Science
Education, Scripting,
Academic Research

Major paradigms

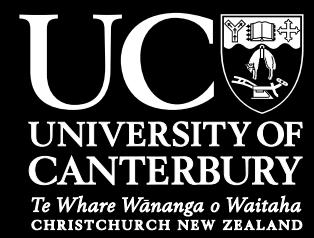
Functional, Dynamic
Typing

Features

Homoiconic,
Minimalistic, Cross
Platform

Basic

<http://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
```

First appeared in

1964

Popular for

Computer Science
Education, Scripting,
Games

Major paradigms

Imperative, Object
Oriented (some variants)

Features

Many Implementations