

# Ruby

---

WPROWADZENIE DO JĘZYKA



**Fundusze Europejskie**  
Wiedza Edukacja Rozwój

**Unia Europejska**  
Europejski Fundusz Społeczny



# Instalacja w systemie Windows

---

Pobieramy instalator ze strony

<https://rubyinstaller.org/downloads/>

Pobieramy **Ruby+Devkit 2.5.5.1 (x64)**

Uruchamiamy instalator i postępujemy zgodnie z wyświetlanymi poleceniami



# Ruby – informacje ogólne

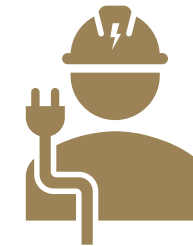
---



Ruby jest językiem skryptowym



Biblioteki w Ruby nazywamy **gemami**



Najczęściej wykorzystywany jest razem z frameworkiem **Rails** do tworzenia serwisów internetowych

# Ruby – słowem wstępu

---

Ruby jest językiem dającym dużą swobodę programiście

Na zapisanie danej operacji zazwyczaj mamy kilka różnych sposobów

Język bardzo mało narzuca – np. możemy używać nawiasów przy wywoływaniu funkcji, ale nie musimy

# Ruby - IDE

---

Jednym z najlepszych IDE do tworzenia aplikacji w języku Ruby jest RubyMine

<https://www.jetbrains.com/ruby/>

Darmowy do użytku edukacyjnego

Aby uzyskać darmowy dostęp do narzędzia, należy założyć konto edukacyjne na portalu JetBrains

<https://www.jetbrains.com/student/>

# Ruby

---

KONSTRUKCJE JĘZYKA

# Komentarze

```
# Komentarz jednolinijkowy
```

```
=begin
```

```
To jest
```

```
komentarz
```

```
w wielu liniijkach
```

```
=end
```

# Zmienne

```
a = 5
```

```
b = 6.5
```

```
str = "Hello World!"
```



# Operatory arytmetyczne

```
a = 10
```

```
b = 3
```

```
suma = a + b           # 13
```

```
roznica = a - b        # 7
```

```
iloczyn = a * b        # 30
```

```
iloraz = a / b         # 3
```

```
iloraz2 = a.to_f / b   # 3.3333...
```

```
modulo = a % b         # 1
```

```
potega = a ** b        # 1000
```

# Operatory przypisania

```
a = 10 # 10
```

```
a += 1 # 11
```

```
a -= 1 # 10
```

```
a *= 2 # 20
```

```
a /= 2 # 10
```

```
a %= 4 # 2
```

```
a **= 3 # 8
```

# Operatory porównania

```
a = 10  
b = 5
```

```
a == b    # false  
a != b    # true
```

```
a < b     # false  
a <= b    # false
```

```
a > b     # true  
a >= b    # true
```

# Operatory porównania c.d.

```
10 <=> 5 # 1  
10 <=> 10 # 0  
10 <=> 15 # -1
```

*Wynik  $a <=> b$ :*

- *1 gdy  $a > b$*
- *0 gdy  $a == b$*
- *-1 gdy  $a < b$*

## Operatory porównania c.d.

```
10.eql?(10) # true  
10.eql?(10.0) # false  
10 == 10.0 # true
```

*Funkcja eql? Porównuje nie tylko wartości, ale także typ.*

# Instrukcja warunkowa - if

```
temperatura = 25

if temperatura < 10
  puts "Zimno!"
elsif temperatura < 20
  puts "Cieplo!"
else
  puts "Goraco!"
end
```

# Instrukcja warunkowa - if

```
blad = true  
puts "Wystapil blad!" if blad
```

# Instrukcja warunkowa - unless

```
temperatura = -5
```

```
unless temperatura > 0  
  puts "Mroz!"  
end
```



Instrukcja  
warunkowa –  
unless

```
blad = false  
puts "Nie ma bledu!" unless blad
```

# Instrukcja warunkowa – case (switch)

```
wiek = 18

case wiek
when 0 .. 2
  puts "niemowle"
when 3 .. 6
  puts "male dziecko"
when 7 .. 12
  puts "dziecko"
when 13 .. 18
  puts "nastolatek"
else
  puts "dorosly"
end
```

# Pętla warunkowa - while

```
i = 0  
n = 5  
  
while i < n do  
  puts "i = #{i}"  
  i += 1  
end
```

# Pętla warunkowa – do while

```
i = 0  
n = 5  
  
begin  
  puts "i = #{i}"  
  i += 1  
end while i < n
```

# Pętla warunkowa until

```
i = 0  
n = 5  
  
until i > n do  
  puts "i = #{i}"  
  i += 1  
end
```

# Pętla warunkowa – do until

```
i = 0  
n = 5  
  
begin  
  puts "i = #{i}"  
  i += 1  
end until i > n
```

Pętla licząca –  
for

Przedział  
zamknięty

```
for i in 0..5  
  puts "i = #{i}"  
end
```

Pętla licząca –  
for

Przedział  
otwarty

```
for i in 0...5  
  puts "i = #{i}"  
end
```



# Pętla iteracyjna - each

```
(0..5).each do |i|  
  puts "i = #{i}"  
end
```

# Tablice

```
tab = Array.new  
puts tab.size      # 0  
puts tab.length   # 0
```

```
tab = Array.new(10)  
puts tab.size     # 10
```

# Tablice - tworzenie

```
tab = Array.new(5, 20)
puts "#{tab}"      # [20, 20, 20, 20, 20]
```

```
tab = Array.new(5) { |e1| e1 = e1 * 2 }
puts "#{tab}"      # [0, 2, 4, 6, 8]
```

```
tab = Array(0..5)
puts "#{tab}"      # [0, 1, 2, 3, 4, 5]
```

```
tab = [1, 2, 3, 4, 5]
puts "#{tab}"      # [1, 2, 3, 4, 5]
```

# Tablice

```
tab = [1, 2, 3, 4, 5]  
puts tab[0]      # 1  
puts tab[1]      # 2
```

# Hashe (Objekty)

```
student = {imie: "Jan", nazwisko: "Kowalski",  
wiek: 22}
```

```
puts student # {:imie=>"Jan",  
:nazwisko=>"Kowalski", :wiek=>22}
```

```
puts student[:imie] # Jan
```

# Funkcje

```
def suma(a, b)  
    a + b  
end
```

```
puts suma(2, 5) # 7
```

# Funkcje

```
def punkt(a)
  x = a
  y = a * 2
  return x, y
end
```

```
puts punkt(5)           # 5 10
```

# Klasy

```
class Prostokat
  attr_accessor :wys
  attr_accessor :szer

  def initialize(wysokosc, szerokosc)
    @wys = wysokosc
    @szer = szerokosc
  end

  def pole
    @wys * @szer
  end

  def kwadrat?
    @wys == @szer
  end
end
```

```
prostokat = Prostokat.new(5, 10)
puts prostokat.pole
puts prostokat.kwadrat?
```

```
prostokat.wys = 10
puts prostokat.kwadrat?
```