

Java þýðing

Editor -> HelloWorld.java

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

Til að þýða í cmd línu

```
% javac HelloWorld.java
```

býr til skrána HelloWorld.class

Til að keyra forritið

```
% java HelloWorld  
Hello World  
%
```

HelloWorld

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

1. `public class HelloWorld {`
 - `public` og `class` verða útskýrð seinna
 - `HelloWorld` nafnið á klasanum, verður að vera í skránni `HelloWorld.java`, há- og lágstafir skipta máli, ekki `helloworld` eða `HelloWorld`
 - { opinn slaufusvigi opnar skilgreininguna á innnyflum klasans
 - Þangað til seinna munum við ekki gera greinarmun á klasa og forriti
2. `public static void main(String[] args) {`
 - `public static void main` er galdrapula
 - `String[] args` eru viðföng forritsins
 - { opinn slaufusvigi opnar innfyli `main` aðferðarinnar
3. `System.out.println("Hello World");`
 - `System.out.println()` er aðferð/fall sem prentar streng út á skjá
 - "Hello World" tákna streng með stafina ...

Inntak – úttak

```
public class UseArgument {
    public static void main(String[] args) {
        System.out.print("Hi, ");
        System.out.print(args[0]);
        System.out.println(". How are you?");
    }
}
```

Prentar örlítið meira en síðast

- args[0] er fyrsta viðfang forritsins
- ATH print() prentar beint út, println() prentar og fer í næstu línu

```
% java UseArgument siggi
Hi siggi. How are you?
%
```

Gagnatög

Gagnatag er mengi gilda og aðgerðir á þessum gildum

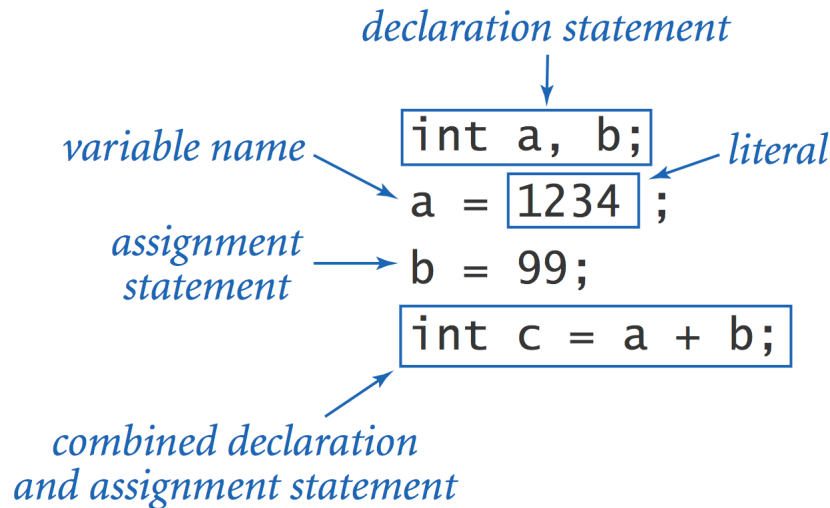
Tag type	Mengi gilda	lesgildi literal value	aðgerðir
char	stafir	'A' '@'	samanburður
String	runa af stöfum	"Hello World" "126 is fun"	samskeyting
int	heiltölur	17 12345	samlagning, margföldun, ...
double	“rauntölur”	3.1415 6.022e23	samglagning, margföldun, ...
boolean	sanngildi	true false	and, or, not

Skilgreiningar

Breyta (variable) er nafn sem vísar til gildis.

Gildingarsetning (assignment statement)

gefur breytu gildi



Tagskipting

Java er “rammlega tagskipt” (strongly typed)

- Öll gildi hafa tög
- Breytur hafa tög
 - Breytur þarf að skilgreina fyrirfram með tagi
 - Ekki hægt að skipta um tag
- Hjálpar til við að skrifa rétt forrit
- heiltölur eru ekki strengir, viljum ekki blanda saman

Arienne 5 sprakk vegna villu með tag á breytu

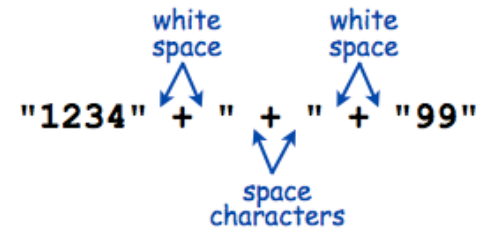
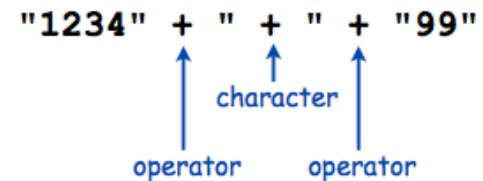


Strengir

String gagnatagið

- Notað fyrir allan texta í Java
- Afmarkaður í forritum með gæsalöppum “ “
- Strengjum skeytt saman með + virkjanum

<i>expression</i>	<i>value</i>
"Hi, " + "Bob"	"Hi, Bob"
"1" + " 2 " + "1"	"1 2 1"
"1234" + " " + " " + "99"	"1234 + 99"
"1234" + "99"	"123499"



Strengir

char gagnatagið

- 2^{16} mögulegir stafir (unicode)
- Lengra en venjulegt ASCII ($2^8 = 256$ stafir)
- Táknaðir með ' ' í forritum
- Sumir stafir eru skrítnir, þarf \ á undan til að tákna
 - '\n' línubil (newline)
 - '\t' tab
 - '\"' einföld gæsalöpp
 - '\"' tvöföld gæsalöpp
 - '\\' öfugt skástrik
- Þurfum stundum að nota í strengjum
 - "Hello \"World\""

Strengir

```
public class UseArguments2 {  
    public static void main(String[] args) {  
        System.out.println("Hi, "+args[0]+", how are you?");  
    }  
}
```

Skrifum út einn streng

- ATH við verðum að setja bil (space) sjálf
- Köllum á forritið með

```
% java UseArguments2 Páll  
Hi, Páll, how are you?  
%
```

```

public class Ruler {
    public static void main(String[] args) {
        String ruler1 = "1";
        String ruler2 = ruler1 + " 2 " + ruler1;
        String ruler3 = ruler2 + " 3 " + ruler2;
        String ruler4 = ruler3 + " 4 " + ruler3;
        System.out.println(ruler4);
    }
}

```

```

        "1"
      "1 2 1"
    "1 2 1 3 1 2 1"

```

```
% java Ruler
```

```
1 2 1 3 1 2 1 4 1 2 1 3 1 2 1
```

```

      |
    | |
  | | |
 | | | |
1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

```

Heiltölur

int gagnatagið er notað til að geyma heiltölur

<i>values</i>	integers between -2^{31} and $+2^{31}-1$				
<i>typical literals</i>	1234	99	-99	0	1000000
<i>operations</i>	add	subtract	multiply	divide	remainder
<i>operators</i>	+	-	*	/	%

<i>expression</i>	<i>value</i>	<i>comment</i>
5 + 3	8	
5 - 3	2	
5 * 3	15	
5 / 3	1	no fractional part
5 % 3	2	remainder
1 / 0		run-time error
3 * 5 - 2	13	* has precedence
3 + 5 / 2	5	/ has precedence
3 - 5 - 2	-4	left associative
(3 - 5) - 2	-4	better style
3 - (5 - 2)	0	unambiguous

ATH

- 5 / 3 er 1, heiltöludeiling
- 5 % 3 er 2, afgangur af deilingu
- enda er $5 = 3*1 + 2$
- röð aðgerða er “venjuleg”

Heiltölur

Heiltölur eru til í nokkrum stærðum

- byte, 8 bitar -128 til 127
- short, 16 bitar -32,768 til 32,767
- int, 32 bitar -2,147,483,648 til 2,147,483,647
- long, 64 bitar -9,223,372,036,854,775,808 til 9,223,372,036,854,775,807

Við munum bara halda okkur við `int` nema
að við þurfum stærri tölur

Heiltöludæmi

```
public class IntOps {  
    public static void main(String[] args) {  
        int a = Integer.parseInt(args[0]);  
        int b = Integer.parseInt(args[1]);  
        int sum  = a + b;  
        int prod = a * b;  
        int quot = a / b;  
        int rem  = a % b;  
        System.out.println(a + " + " + b + " = " + sum);  
        System.out.println(a + " * " + b + " = " + prod);  
        System.out.println(a + " / " + b + " = " + quot);  
        System.out.println(a + " % " + b + " = " + rem);  
    }  
}
```

```
% javac IntOps.java  
% java IntOps 1234 99  
1234 + 99 = 1333  
1234 * 99 = 122166  
1234 / 99 = 12  
1234 % 99 = 46
```

Fleytitölur

double gagnatagið er notað fyrir “rauntölur”

<i>values</i>	real numbers (specified by IEEE 754 standard)				
<i>typical literals</i>	3.14159	6.022e23	-3.0	2.0	1.4142135623730951
<i>operations</i>	add	subtract	multiply	divide	
<i>operators</i>	+	-	*	/	

<i>expression</i>	<i>value</i>
3.141 + .03	3.171
3.141 - .03	3.111
6.02e23 / 2.0	3.01e23
5.0 / 3.0	1.6666666666666667
10.0 % 3.141	0.577
1.0 / 0.0	Infinity
Math.sqrt(2.0)	1.4142135623730951
Math.sqrt(-1.0)	NaN

ATH

- 5 / 3 er 1, heiltöludeiling
- 5.0 / 3.0 er 1.66... fleytitöludeiling
- 1 / 0 er villa!
- 1.0 / 0.0 er talan “óendanlegt”

Java Math

```
public class Math
```

```
double abs(double a)           absolute value of a
```

```
double max(double a, double b) maximum of a and b
```

```
double min(double a, double b) minimum of a and b
```

Note 1: abs(), max(), and min() are defined also for int, long, and float.

```
double sin(double theta)       sine function
```

```
double cos(double theta)       cosine function
```

```
double tan(double theta)       tangent function
```

Note 2: Angles are expressed in radians. Use toDegrees() and toRadians() to convert.

Note 3: Use asin(), acos(), and atan() for inverse functions.

```
double exp(double a)           exponential ( $e^a$ )
```

```
double log(double a)           natural log ( $\log_e a$ , or  $\ln a$ )
```

```
double pow(double a, double b) raise a to the bth power ( $a^b$ )
```

```
long round(double a)           round to the nearest integer
```

```
double random()                random number in [0, 1)
```

```
double sqrt(double a)          square root of a
```

```
double E                       value of e (constant)
```

```
double PI                      value of  $\pi$  (constant)
```

Dæmi: Lausn 2. stigs jöfnu

$$x^2+bx+c = 0$$

$$x = \frac{-b \pm \sqrt{b^2 - 4c}}{2}$$

```
public class Quadratic {
    public static void main(String[] args) {

        // parse coefficients from command-line
        double b = Double.parseDouble(args[0]);
        double c = Double.parseDouble(args[1]);

        // calculate roots
        double discriminant = b*b - 4.0*c;
        double d = Math.sqrt(discriminant);
        double root1 = (-b + d) / 2.0;
        double root2 = (-b - d) / 2.0;

        // print them out
        System.out.println(root1);
        System.out.println(root2);
    }
}
```