

|  |  |
| --- | --- |
| *s*.length() *s*.charAt(*i*) | length of *s*  extract *i*th character |

**Java Statements (cont)**

switch ( *expre*​*ssion* ) { case *value*:

*st*​*ate*​*ments*

break; case *value2*:

*st*​*ate*​*ments*

break; default:

*st*​*ate*​*ments*

}

**Exception Handling**

try { statements;

} catch (*Exce*​*pti*​*onType e1*) { statements;

} catch (Exception *e2*) { catch-all statements;

} finally { statements;

}

**Java Data Conversions**

**String to Number**

int i = Integer.parseInt(​*str*);

double d = Double.parseDouble(*s*​*tr*);

**Any Type to String**

String s = String.valueOf(*va*​*lue*);

**Numeric Conversions**

int i = (int) *numeric expres*​*sion*;

**Java String Methods**

*s*.substring(​*start*, *end*)

*s*.toUpperCase()

substring from *start* to *end*-1

returns copy of *s* in ALL CAPS

returns copy of *s* in lowercase

index of first occurence of

*x*

*s*.toLowerCase()

*s*.indexOf(*x*)

search and replace

|  |  |
| --- | --- |
| *s*.split(*r*​*egex*) *s*.trim() | splits string into tokens trims surrounding whitespace  true if s equals s2  0 if equal/+ if s > s2/- if s < s2 |
|  |
| *s*.equals(*s2*) *s*.compareTo(*s2*) |
|  |

.

**java.util.ArrayList Methods**

|  |  |
| --- | --- |
| *l*.add(​*itm*) | Add *itm* to list |
| *l*.get(*i*) | Return *i*th item |
| *l*.size() | Return number of items |
| *l*.remove(*i*) | Remove *i*th item |
| *l*.set(*i*, *val*) | Put *val* at position *i* |

ArrayList<String> names = new ArrayList<String>();

*s*.replace(*old*, *new*)

**Java Data Types**

byte / short / int / long -123, 10

|  |  |
| --- | --- |
| float / double char boolean String | 235.13  'U'  true, false "Greetings from earth" |

**Java Statements**

**If Statement**

if ( *expre*​*ssion* ) { *st*​*ate*​*ments*

} else if ( *expre*​*ssion* ) { *st*​*ate*​*ments*

} else {

*st*​*ate*​*ments*

}

**While Loop**

while ( *expre*​*ssion* ) { *st*​*ate*​*ments*

}

**Do-While Loop**

do {

*st*​*ate*​*ments*

} while ( *expre*​*ssion* );

**For Loop**

for ( int i = 0; i < *max*; ++i) { *st*​*ate*​*ments*

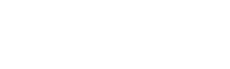
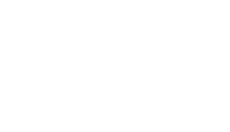
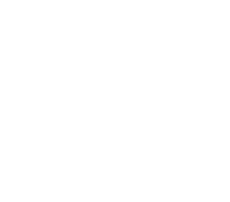
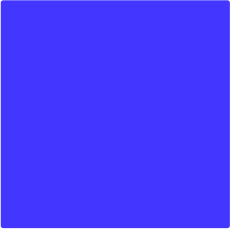
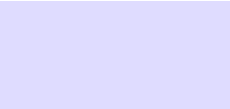
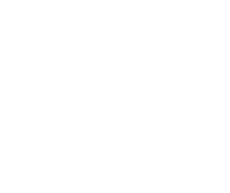
}

**For Each Loop**

for ( *var* : *colle*​*ction* ) { *st*​*ate*​*ments*

}

**Switch Statement**



**Java Boolean Operators**

! x (not) x && y (and) x || y (or)

true if contains *key*

**Java Text Formatting**

**printf style formatting**

System.out.printf("Count is %d\n", count); s = String.format("Count is %d", count);

**MessageFormat style formatting**

s = MessageFormat.format( "At {1,time}, {0} eggs hatched.", 25, new Date());

**Individual Numbers and Dates**

s = NumberFormat.getCurrencyInstance() .format(x);

s = new SimpleDateFormat(""h:mm a"") .format(new Date());

s = new DecimalFormat("#,##0.00") .format(125.32);



**java.util.HashMap Methods**

|  |  |
| --- | --- |
| *m*.put(​*key*​,​*value*) | Inserts *value* with *key* |
| *m*.get(​*key*) *m*.containsKey(*key* ) | Retrieves value with *key* |

HashMap<StÂrinÂg,String> names = new HashMap<StÂrinÂg, String>();

See

http://docs.oracle.com/javase/6/docs/api/java/util/ HashMap.html for more.

**Java Hello World**

import java.util.Date; public class Hello {

public static void main(String[] args) { System.out.println("Hello, world!"); Date now = new Date(); System.out.println("Time: " + now);

} }

\*Save in Hello.java

\* Compile: **javac Hello.java**

\* Run: **java Hello**

**Java Arithmetic Operators**

|  |  |  |  |
| --- | --- | --- | --- |
| x + y x \* y x % y | add multiply modulus | x - y  x / y ++x / x++ --x / x-- | subtract divide increment decrement |

Assignment shortcuts: x *op*= y Example: x += 1 increments x

**Java Comparison Operators**

|  |  |  |  |
| --- | --- | --- | --- |
| x < y | Less | x <= y | Less or eq |
| x > y | Greater | x >= y | Greater or eq |
| x == y | Equal | x != y | Not equal |