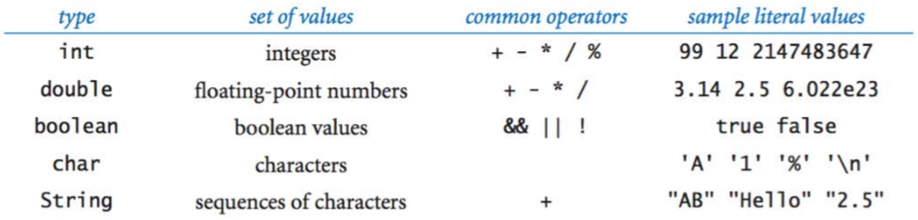
Java 8 CHEAT SHEET

Hello World

```java
public class HelloWorld {

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

Some built-in data types



type	*set of values*	*common operators*	*sample literal values*
int	integers	+ - * / %	99 12 2147483647
double	floating-point numbers	+ - * /	3.14 2.5 6.022e23
boolean	boolean values	&& || !	true false
char	characters		'A' '1' '%' '\n'
String	sequences of characters	+	"AB" "Hello" "2.5"

Classes and Objects

```java
public class Person {
    private int age;
    private String name;

    public Person(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    @Override
    public String toString() {
        return name;
    }
}

public class PersonUsage {
    public static void main(String[] args) {
        Person willie = new Person("Willie Wortel");
        willie.setAge(40);
        int value = willie.getAge();
    }
}
```

Operators

Arithmetic Operators	Assignment Operators	Logical Operators	Relational Operators
+ (Addition)	=	&& (logical and)	== (equal to)
- (Subtraction)	+=	|| (logical or)	!= (not equal to)
* (Multiplication)	-=	! (logical not)	> (greater than)
/ (Division)	*=		< (less than)
% (Modulus)	/=		>= (greater than or equal to)
++ (Increment)	%=		<= (less than or equal to)
-- (Decrement)			

//	Comment to end of line
/* x */	Comment everything between
/** x */	Javadoc comment
public	Can be seen anywhere in any package
private	Can be seen within class
protected	In package and subclasses of this in any package
static	Shared by all instances of a class
final static	Constant

Strings

`boolean`	`equals(Object anObject)` Compares this string to the specified object.
`boolean`	`equalsIgnoreCase(String anotherString)` Compares this `String` to another `String`, ignoring case considerations.
`int`	`length()` Returns the length of this string.
`String[]`	`split(String regex)` Splits this string around matches of the given regular expression.

Inheritance

```
public class Swimmer extends Person {
    private double personalBest;

    public Swimmer(String name, double personalBest) {
        super(name);
        this.personalBest = personalBest;
    }
}
```

Interface

```
public interface Studying {
    void study();
}

public class Person interface Studying {
```

Polymorphism

```
Person someone = new Swimmer("Michael Phelps", 50.77);
```

Input

```
Scanner input = new Scanner(System.in);
int number = input.nextInt();
input.nextLine();
String line = input.nextLine();
input.close();
```

Arrays

```
int[] a = new int[50];
a[0] = 5;
a[a.length - 1] = 5;
for (int i = 0; i < a.length; i++) {
    System.out.println(a[i]);
}
```

LocalDate and LocalDateTime

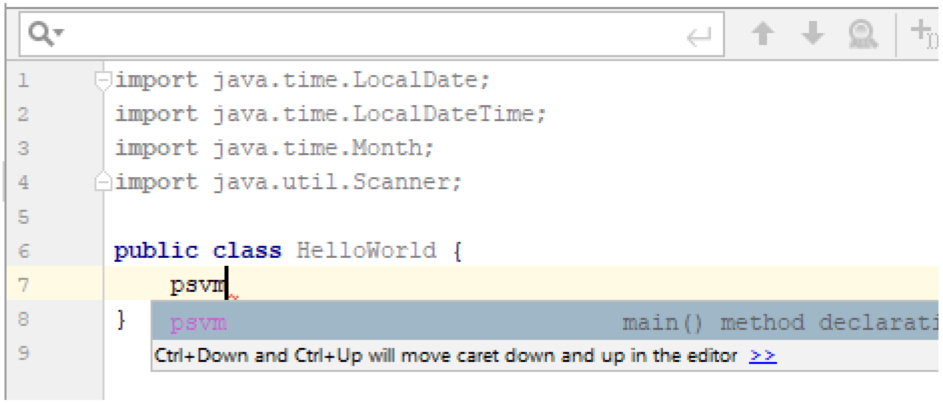
```
LocalDate birthDay = LocalDate.of(1990, Month.DECEMBER,
15);
LocalDateTime fiveHoursBefore =
LocalDateTime.now().minusHours(5);
```

Exception handling

```
try {
    // some code that can throw an exception
} catch (Exception e) {
    // exception handling
} finally {
    // some code executed with or without error
}
```

IntelliJ

, Ctrl + Shift + A	Search through all commands in IntelliJ
Ctrl + Alt + L	Reformat code and organize imports

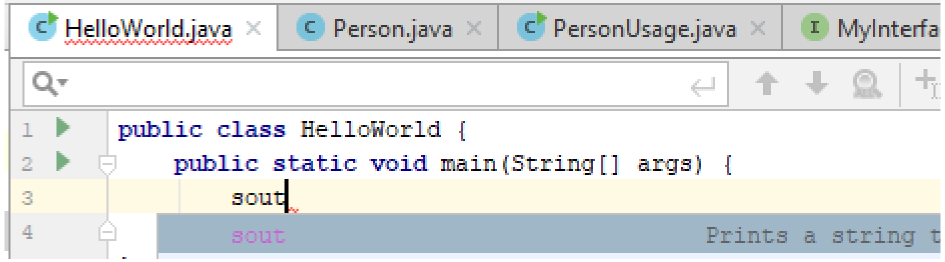


```
import java.time.LocalDate;
import java.time.LocalDateTime;
import java.time.Month;
import java.util.Scanner;

public class HelloWorld {
    psvm
}
```

```
psvm                              main() method declarati
```

Ctrl+Down and Ctrl+Up will move caret down and up in the editor >>



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

```
sout                              Prints a string t
```