

Java Types

Primitive Types

- int
- boolean
- double

Class Types

- String
- Picture
- Pixel
- Circle

Array Types

- <base-type> []
- int[]
- double[]
- String[]
- Pixel[]

Java Expressions

have a Type and a Value

In general, are composed of

- constant

- variable

 - local scope variable

 - instance field variable

 - Class variable

- Class method returning value

- Object method returning value

- new <object-or-array-creation>

- <sub-expression>

all put together with operands

Java Statements

- Declaration statements

- Assignment statement

- return statement

- method definition statement

- class definition statement

- if-statement

- for-statement

- while-statement

Today

- Complete Grayscale example

- Cover conditionals (if-statement)

- Pairs working session on example Java code

Conditionals

Allows us to tell the computer to make decisions based on the current state of the program -- what values instance fields have, what values parameters to methods have, etc.

We are no longer constrained to do exactly the same sequence of steps every time.

sequence of steps every time.

We will discuss three forms. There are more subtle variations, but these three will be sufficient for now.

Form 1: The simple if statement.

General syntax:

```
if (<boolean-expression>
    <statement>
```

As before, we substitute for <boolean-expression> and for <statement>. Most often, <statement> is a block statement, so that we can group together a set of statements and treat them as a unit.

Example:

```
if (age < 21)
{
    entry = false;
    System.out.println("You may not come in");
}
```

General semantics:

1. Evaluate <boolean-expression>, yielding either true or false.
2. If <boolean-expression> yielded true, execute <statement> and then continue executing whatever follows, otherwise, simply skip over <statement> and continue executing whatever follows <statement>.

Form 2: The if-else statement.

General syntax:

```
if (<boolean-expression>
    <statement1>
else
    <statement2>
```

As before, we substitute for <boolean-expression> and for <statement1> and <statement2>. Most often, these are block statements, so that we can group together a set of statements and treat them as a unit.

Example:

```
if (age < 21) {
```

```

if (age < 21) {
    entry = false;
    System.out.println("You may not come in");
} else {
    entry = true;
    System.out.println("Come right on in");
}

```

General semantics:

3. Evaluate <boolean-expression>, yielding either true or false.
4. If <boolean-expression> yielded true, execute <statement> and then continue executing whatever follows, otherwise, simply skip over <statement> and continue executing whatever follows <statement>.

Form 3: The if-else-if statement.

General syntax:

```

if (<boolean-expression1>)
    <statement1>
else if (<boolean-expression2>)
    <statement2>
else if (<boolean-expression3>)
    <statement3>
else
    <statement-else>

```

Example:

```

if (ages < 10) {
    System.out.println ("child");
} else if (age >=10 && age < 13) {
    System.out.println ("pre-teen");
} else if (age >=13 && age < 20) {
    System.out.println ("teenager");
} else {
    System.out.println ("adult");
}

```