Using library methods

To do math:

1. `Math.min(5,12);`
2. `Math.abs(-27);`

To get user input:

1. `Scanner scnr = new Scanner(System.in);`
2. `int numPages = scnr.nextInt();`

Note for `Scanner`: check your inputs! `scnr.hasNextInt();`

$\Rightarrow$ `Math` javadoc vs `Scanner` javadoc

Strings and characters

`String` literal: `"string"`
`char` literal: `'c'`

Empty `String`: `""`
Adding `Strings` together: `"Hello,"+" world!"`

String operations

With `Scanner`:

- Getting a `char`: `in.next();`
- Getting a `String`: `in.nextLine();`

Using `String` methods:

- To get a particular character: `s.charAt(ind);`
- To get a substring: `s.substring(beg,end);`
- To find the length: `s.length();`

Note: 0-based indexing!

Random Number Generators

In Java: `Random` class

$\Rightarrow$ Must include line `import java.util.Random;`

To declare: `Random rng = new Random();`

To get a number: `int rand = rng.nextInt(range);`

Program Flow

Up through now, program flow has been *sequential*:

$$a \rightarrow b \rightarrow c \rightarrow d \rightarrow ...$$

But we can have non-sequential flow in programs!
$\Rightarrow$ `if`, `if-else`, `if-else-if`

Relational operators


```
if (<condition>) {...}
else if (<condition>) {...}
```

What are these conditions? $\Rightarrow$ *boolean* conditions (true/false)

Use: `==`, `!=`, `<`, `<=`, `>`, `>=`, `string.equals()`

NOT: `x = y` or `x =< y`

booleans

Also can have `boolean`-type variables/literals:

```
boolean agree = (x == y);
if(agree) {...}
agree = true;
```

Practice

With a partner: write a dice rolling program!

1. Prompt user for number of sides on the die
2. Get user input
3. Generate random number as a roll
4. Output result of roll
5. Output "even" if the roll is even, "odd" if the roll is odd