

User Input

Why User Entry?

> When we assign values (age = 15), we are making a program **STATIC** ... it never changes.

> If we ask the user to enter the values, it becomes **DYNAMIC** ... it changes depending on who the user is that runs the program

User-entry has the following SYNTAX:

```
identifier = c.readTYPE();
```

- > identifier must be defined as the type of data which is expected
- > TYPE can be Int, Double, Char, String, Line (Note the Cap 1st letter)

User Input

Examples:

Integer Entry:

```
c.print ("Please enter your age: "); // prompt -entry on same line
age = c.readInt(); // prog. waits for user entry
c.println ("Your age is " + age); // display of data
```

Double Entry:

```
c.print ("Please enter your wage:"); // prompt with entry same line
wage = c.readDouble(); // prog. waits for user entry
c.print ("You are making $"); // output (no + and , together)
c.print(wage,5,2); // displaying with field descriptive
c.println ("/hour."); // ending sentence with literal
```

** integer and double are pretty straight forward

User Input

Examples Con't:

String Entry:

```
c.print ("Please enter your name: ");    // prompt -entry on same line
name = c.readLine();                  // prog. waits for user entry
c.println ( "Your name is " + name);    // display of data
```

What is the difference between `c.readLine();` and `c.readString();`?

```
c.readLine();           -remembers ALL you type, including spaces
c.readString();        -only remembers all information prior to a space or
                        up to a <enter>
```

For example:

If your entry is PETER PAN for each of the following:

```
name1 = c.readLine();   has name1 holding PETER PAN
name2 = c.readString(); has name2 holding PETER
```

User Input

Examples Con't:

Character Entry:

```
c.print ("Please enter A, B, or C: ");      // prompt - entry on same line
letter = c.readChar();                    // prog. waits for user entry
c.println ( "You entered " + letter);     // display of data
```

There is one problem with `c.readChar()` ...

- > every keyboard has a 'buffer' that remembers the keys you hit
- > the `c.readChar()`; reads from the keyboard buffer
- > BUT ... sometimes there is residual data in the buffer, so the `c.readChar()` doesn't seem to work well if used in repetition

Solution:

```
String letter;
c.print ("Please enter A, B, or C: ");
letter = c.readString();
c.println ( "You entered " + letter[0]); // 'subscript' position 0
```