

# Strings

## Part One

# The Chaos Game Revisited

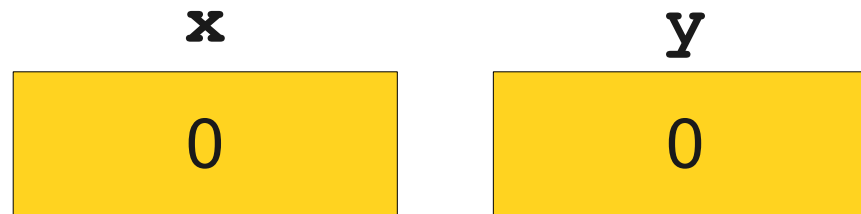
# The Chaos Game

- Pick any three points.
- Starting at any of the points:
  - Choose one of the three points randomly.
  - Move halfway from your current location to the chosen point.
  - Draw a dot at your current location.
  - Repeat.

# A Minor Change

What Just Happened?

```
double x = 0;  
double y = 0;  
  
while (true) {  
    moveRandomly(x, y);  
    plotPixel(x, y);  
}
```



```
GPoint d = getRandomPoint();
```

```
x = (x + d.getX()) / 2.0;
```

```
y = (y + d.getY()) / 2.0;
```

**x**

0

**y**

0

```
GPoint d = getRandomPoint();
```

```
x = (x + d.getX()) / 2.0;
```

```
y = (y + d.getY()) / 2.0;
```

**x**

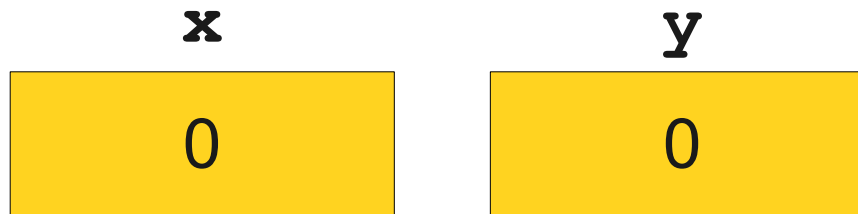
137

**y**

42



```
double x = 0;  
double y = 0;  
  
while (true) {  
    moveRandomly(x, y);  
    plotPixel(x, y);  
}
```

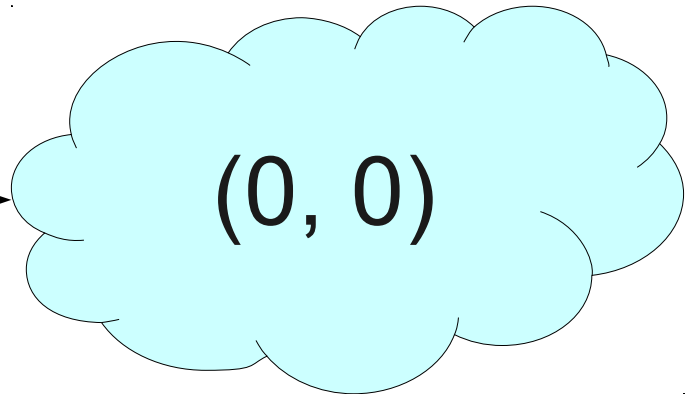
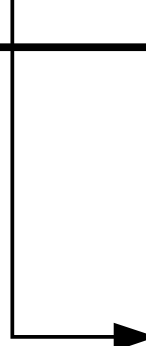
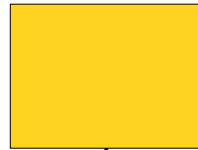




```
GPoint pt = new GPoint(0, 0);
```

```
while (true) {  
    moveRandomly(pt);  
    plotPixel(pt);  
}
```

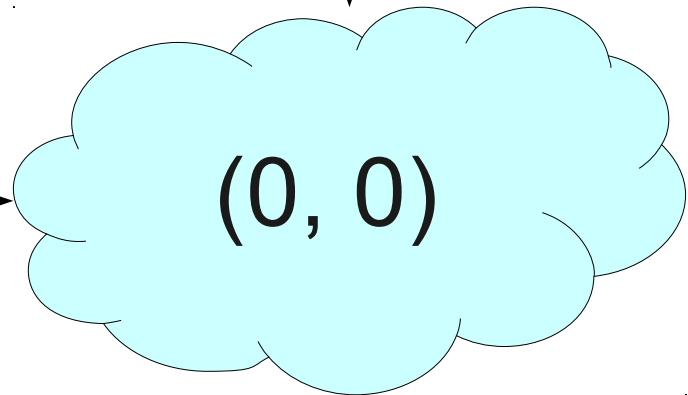
pt



```
GPoint d = chooseRandomPoint();
```

```
pt.setLocation((pt.getX() + d.getX()) / 2.0,  
              (pt.getY() + d.getY()) / 2.0);
```

pt



```
GPoint d = chooseRandomPoint();
```

```
pt.setLocation((pt.getX() + d.getX()) / 2.0,  
              (pt.getY() + d.getY()) / 2.0);
```

pt

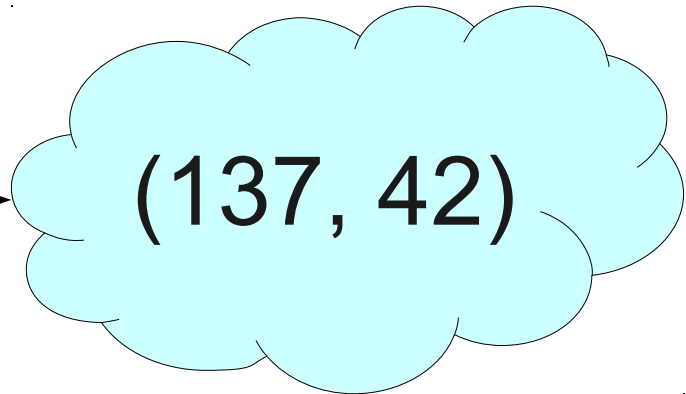
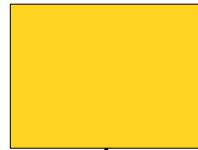


(137, 42)

```
GPoint pt = new GPoint(0, 0);
```

```
while (true) {  
    moveRandomly(pt);  
    plotPixel(pt);  
}
```

pt



# Parameter Passing

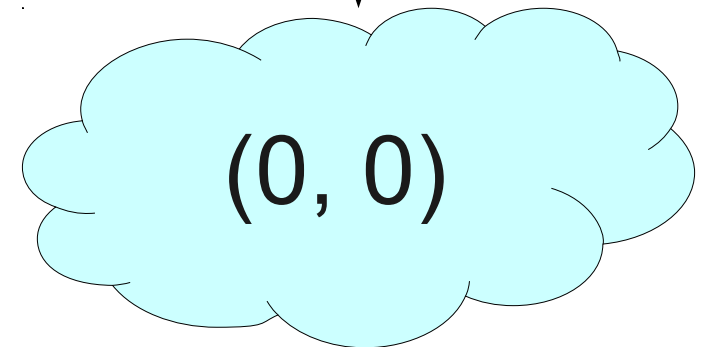
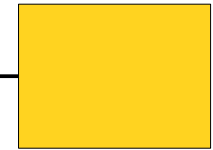
- All parameters in Java are passed by value.
- In Java, variables of primitive type (**int**, **double**, etc.) store actual values.
- In Java, variables of *object* type (G0val, GRect, etc.) don't actually store those objects. They store *references* to those objects.
  - They “point” to where the object really is.

# Another Variation



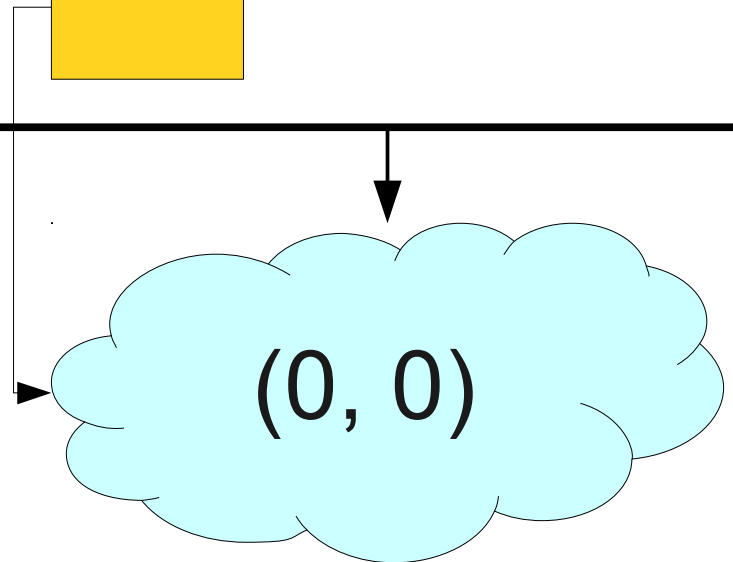
```
GPoint pt = new GPoint(0, 0);  
  
while (true) {  
    moveRandomly(pt);  
    plotPixel(pt);  
}
```

pt



```
GPoint d = chooseRandomPoint();  
GPoint result =  
    new GPoint((pt.getX() + d.getX()) / 2.0,  
              (pt.getY() + d.getY()) / 2.0);  
  
pt = result;
```

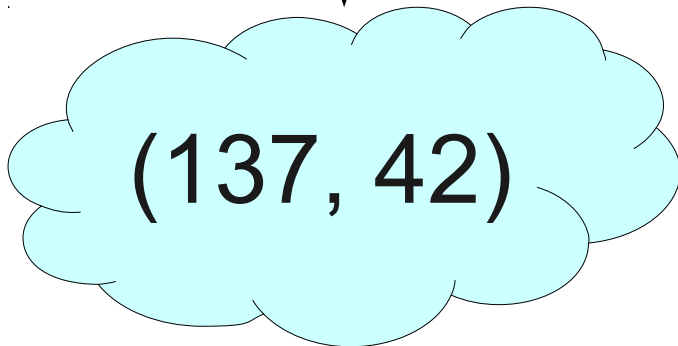
pt



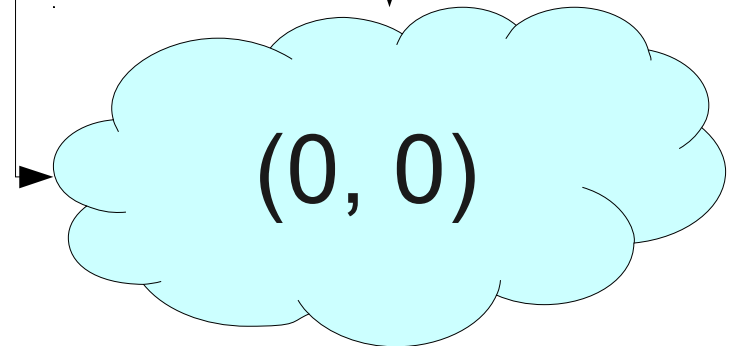
```
GPoint d = chooseRandomPoint();  
GPoint result =  
    new GPoint((pt.getX() + d.getX()) / 2.0,  
               (pt.getY() + d.getY()) / 2.0);
```

```
pt = result;
```

result



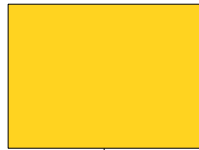
pt



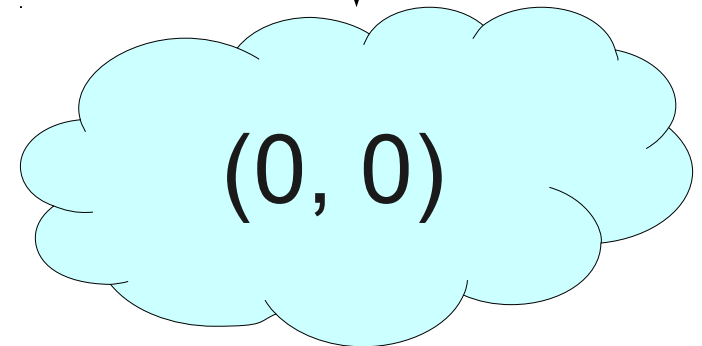
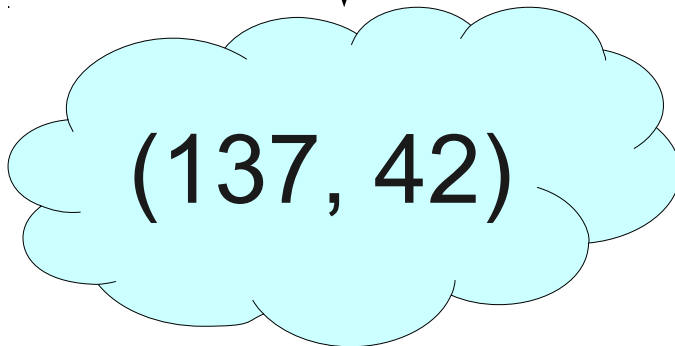
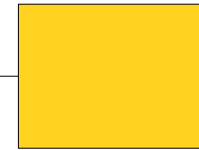
```
GPoint d = chooseRandomPoint();  
GPoint result =  
    new GPoint((pt.getX() + d.getX()) / 2.0,  
               (pt.getY() + d.getY()) / 2.0);
```

```
pt = result;
```

result

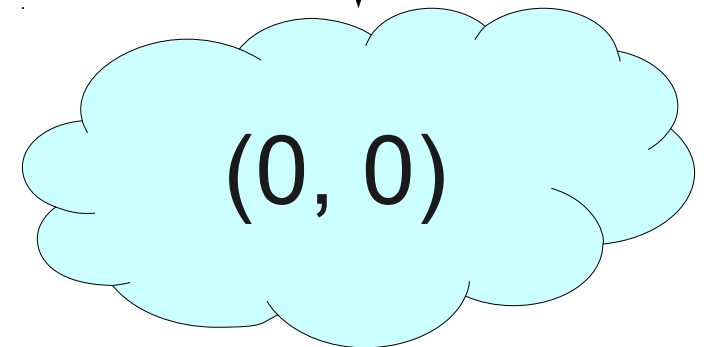
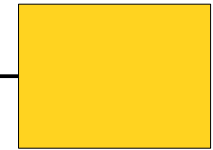


pt



```
GPoint pt = new GPoint(0, 0);  
  
while (true) {  
    moveRandomly(pt);  
    plotPixel(pt);  
}
```

pt



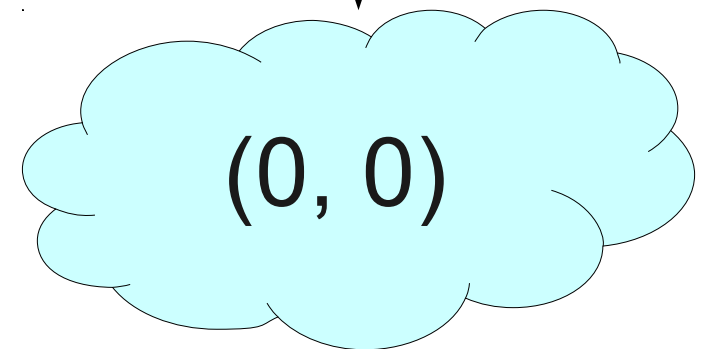
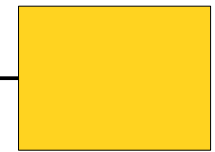
# A Nuance

- If you pass an object into a method, that method can change properties of the object passed in.
  - The caller can then see these changes.
- If you pass an object into a method, that method cannot change ***which object*** is being referred to.
  - The caller will always end up referring to the same object, though the properties of that object might have changed.

One Final Approach...

```
GPoint pt = new GPoint(0, 0);  
  
while (true) {  
    pt = moveRandomly(pt);  
    plotPixel(pt);  
}
```

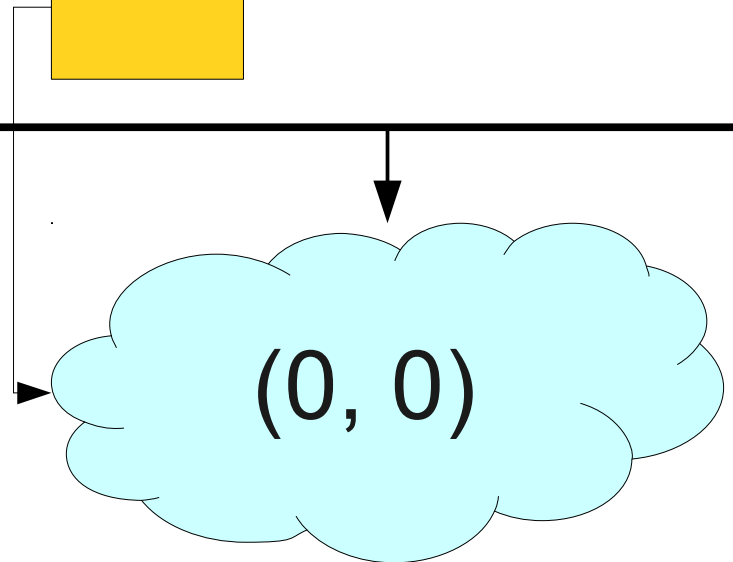
pt





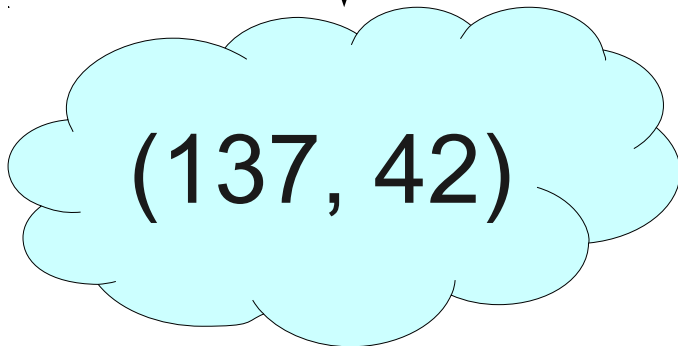
```
GPoint d = chooseRandomPoint();  
GPoint result =  
    new GPoint((pt.getX() + d.getX()) / 2.0,  
               (pt.getY() + d.getY()) / 2.0);  
  
return result;
```

pt

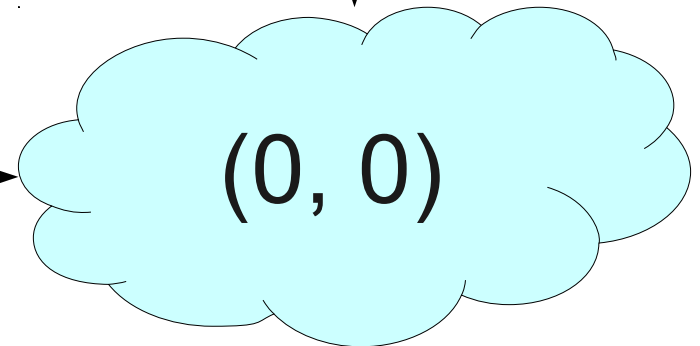


```
GPoint d = chooseRandomPoint();  
GPoint result =  
    new GPoint((pt.getX() + d.getX()) / 2.0,  
              (pt.getY() + d.getY()) / 2.0);  
  
return result;
```

result



pt



```
GPoint pt = new GPoint(0, 0);  
  
while (true) {  
    pt = moveRandomly(pt);  
    plotPixel(pt);  
}
```

pt



(137, 42)

(0, 0)

# Summary

- Primitive types are passed by value.
  - The callee gets a *copy* of the value.
  - The callee can change that *copy*, but cannot change the original.
- Object references are passed by value.
  - The callee gets a copy of the *reference*, not a copy of the *object*.
  - The callee can change the object, but cannot change *which* object is referred to.

**Time-Out for Announcements!**

# Assignment 3

- Assignment 3 is out now, due next Monday at 3:15PM.
- **Recommendation:** Try to get the bricks created and the paddle set up and moving by this Wednesday.

# Midterm Logistics

- First midterm is **Wednesday, February 12** from 7PM - 10PM.
  - Room assignments TBA.
- Open-book, open-note, closed everything else.
- Covers material up through and including strings.
- Practice exam released; solutions will follow later this week.
  - ***Take this practice exam under realistic conditions!***

Back to CS106A!



# An Interesting Article

“How Revolutionary Tools  
Cracked a 1700s Code”

<http://www.nytimes.com/2011/10/25/science/25code.html>

|oghnamōrāuvzīgriqmā|érzueenjyra=rzrphmhmzy|úlk  
 zūfjirzīthiāurhix|nōmūxēlōd+|+uīūst~~at~~x+wipen|nirpī  
 ππi|nxiōuāthpūczpījōdāthjēnū|ōphīrāmōzūh|ōtēnō=rzr  
 hē|zūn:scō|fπīzē. Anzuvōimzūpīrāurzh|w. A|fjōcīp  
 cōduozpīnū|g|pūgōpōdētōdōcīthēdē|laby+pcē|u|f|x|z|g|t|z|y|p|h  
 =r|j|ā|b|h|f|:zē|l|s|ū|y|x|u|e|z|p|g|:z|c|ū|x|g|m|u|z|p|r|=+p|p|ō|z|x|h|g|p|y  
 πīrāfzīnāgrēthāsthīr|z|h|z|=s|z|ut. Δ|j|n|n|l|x|g|z|n|d|j|n|th|l|o|e  
 • pūoogziō|ū+cñx|x|k|mic:k|y|p|o|+|r|p|h|r|z|l|s|l|c  
 • x|n|z|f|s|=h|p|n|h|d|u|d|s|ā|l|u|b|+|z|h|r|p|z|g|s|x|d|=r|l|ū|j|t|ū|z|ō|g|u|b|h|h|c|u  
 • x|n|z|x|c|d|r|l|u|v. Δ|l|c|ē|n|h|z|ū|z|h|n:ō|j|me.  
 z|p|r|v|p|z|y|d|z|ē|r|z|ā. Δ|b|c|p|l|f|h|h|+|d|ō|v|z|h|n:w|p|u|r|x|p|ic|:u  
 c=9|p|b|c|ē|h|j|c|ē|r|x|n|+|=)d|f|p|p|o|=r|z|b|g|h|r|z|ū|+|l|p|ē|r|n|z|j|ū|+|d|n|u|Δ  
 f|p|l|t|z|ē|g|t|t|p|=m|p|z|z|n|k|r|x|ā|b. Δ|ō|j|n|l|c|x|m|z|f|p|u|g. Δ|h|=  
 l|h|p|d|ē|j|p|y|h|ō|l|a|t|+|p|d|ē|l|s|z|ā|z|p|y|j|h|d|z|u|z|ō. Δ|z|u|p|ō|h|g|z|p|p|ē  
 l|u|v|z|n|+|h|f|d|c|:ī|r|:l|d|z|z|h|=|u|+|l|x|z|u|b|m|ō|r|l|u|f:  
 • π|p|z|ū|j|c|ē|=m|ā|l|ū|n|p|r|x|z|z|b|d|c|:n|z|h|ē|ō|j|m|s|p|y|p|d|g|h|ō  
 • +h|c|g|h|g|z|ē|m|p|ē|c|s|=g|ē|h|z|u|h|p|ō|j|f|c|p|c|ī|r|u|w. Δ|o|=m|z|h|m|l  
 x|h|r

f p l t c r

• z|h|r|ā|z|p|n|=l|r|y|p|l|h|d|u|c. Δ|f|h|r|x|x|p|k|z|=p|u|e|ō|h|ū|j|ō|f|d|u|b|d|ō|r|h  
 • p|z|l|h|f|ō|m|p|p|f|r|d|ō|g|l|=m|z|l|h|g|u|h|r|t|z|p|z|ā|x|j|ō|h|l|f|l|:p|g|ē|z|p|r|:  
 • z|v|n|d|ō|r|g|ū|g|u|p|m|h|h|r|u|d. Δ|x|p|ā|z|g|c|p|c|h|p|u|c. Δ|z|x|d|ū|r|:u  
 • b|+|q|:ē|c.  
 V|z|ā|x|t|p|l|:m|ū|h|p|ū|c|ā|h|z|ē|l|g|h|h|d|n|c|z|ō|h|:p|y|r|g|z|g|u|f|d|p|m|:  
 s|ā|r|u|f|z|ē|r|w. Δ|c|=m|p|v|z|z. Δ|x|h|p|l|h|h|t|+|r|z|h|f|m|p|h|z|ē|h|Δ|i  
 n|ā|c|p|ū|b|y|f|r|p|h|n|h|y|f|ō|z|ū|m|d|a|g|h|z|ū|+|g|=z|d|r|x|r|d|s|z|ā|j|g. Δ|o|o  
 p|r|d|c|q|ē|l|b|z|h|k|f|r|g|p|h|+|f|ā|l|z|=|g|x|g|l|r|u|d|x|ā|h|ū|y|+|u|p|r|a|z|l|l|l|l|y  
 ō|r|u|a|m|ō|h|z|=p|p|u|g|m|n|h|z|u|b|d|ā|j|p|g|z|f|p|l|n|h|ē|x|ū|c|l|a|=n|z|p|d|h|z  
 p|h|h|h|t|+|r|z|h|ā|x|p|r|c|p|x|p|y|f|u|ē|m|g|=g|p|v|r|z|ē|z|g|s|=b|ō|h|z|p|r|g|h|p  
 z|r|l|ē|h|t|l|:z|ū|f|ō|l|p|l|z|=s|x|s|y|p|g|u|m|z|m|t|l|p|h|t|+|d|h|ā|p|g|h|c  
 c|u|d|r|m|ū|l|u|z|u|c|x|c|p|d|x|r|p|z|z|ō|j|f|z|z|h|d|y|ē|j|u|z|ā. Δ|n|p|t|c|y|f  
 f|l|n|z|r|+|h|g|z|z|c|Δ|m.  
 P|m|ā|o|o|l|z|m|l|h|=ē|b  
 d|ē|r|:s|y|p|u|d|r|ū|n|p|l|v. Δ.  
 H|ō|l|g|g-Δ|b|m|z|z|f|m|h|p|l|c|d|m|ō|h|c|p|h|z|=j|p|c|ū|r|d|p|r|:l|o  
 z|h|p|r|x|g|=n|p|z|ē|d|t|y|f|j|ā|b|l|s|l|l|d|m|c|p|u|g|x|r|z|t|r|ē|h|p|u|c|d|n|:ā|l|ō|g

A **string** is a sequence of characters.





H e l l o !

H	e	l	l	o	!
---	---	---	---	---	---

0

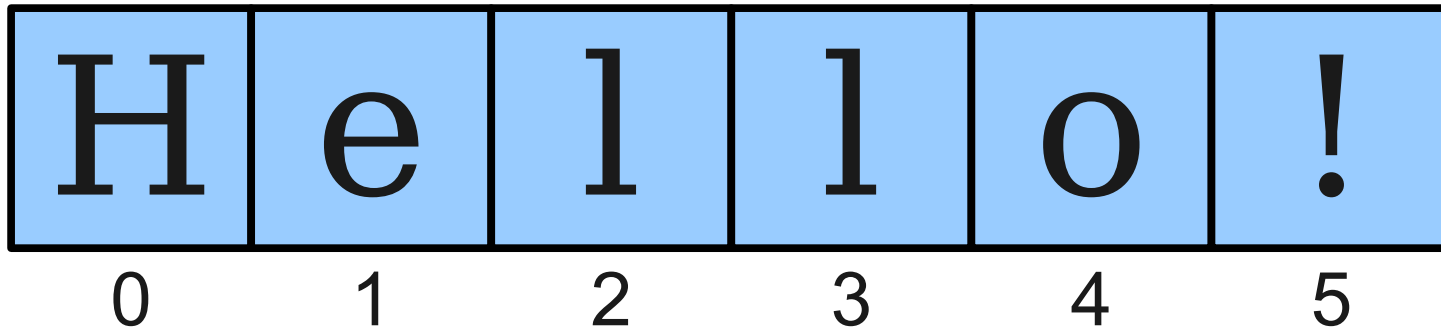
1

2

3

4

5



***string***.charAt (***index***)



# The Data Type **char**

- The primitive type **char** represents a single character or glyph.
- Some examples:

```
char letterA = 'A';
```

```
char plus = '+'
```

```
char zero = '0';
```