

CharAt Loop

Processing Each Letter



```
public class loopChar {  
    public static void main (String args[]) {  
        new loopChar ();  
    }  
}
```

What is the name
of the
constructor?

```
public loopChar () {  
    String s = "word";  
    System.out.println (reverse (s));  
}
```

What is the name
of the method?

```
public String reverse (String s){  
    String ans = "";  
    for (int i = s.length () - 1 ; i >= 0 ; i--)  
        ans += s.charAt (i);  
    return ans;  
}
```

What is
the return
type?

What is the
parameter?

```
}
```

```
public class loopChar {  
    public static void main (String args[]) {  
        new loopChar ();  
    }  
}
```

Where is the
method called?

```
public loopChar () {  
    String s = "word";  
    System.out.println (reverse (s));  
}
```

What is passed
into the method?

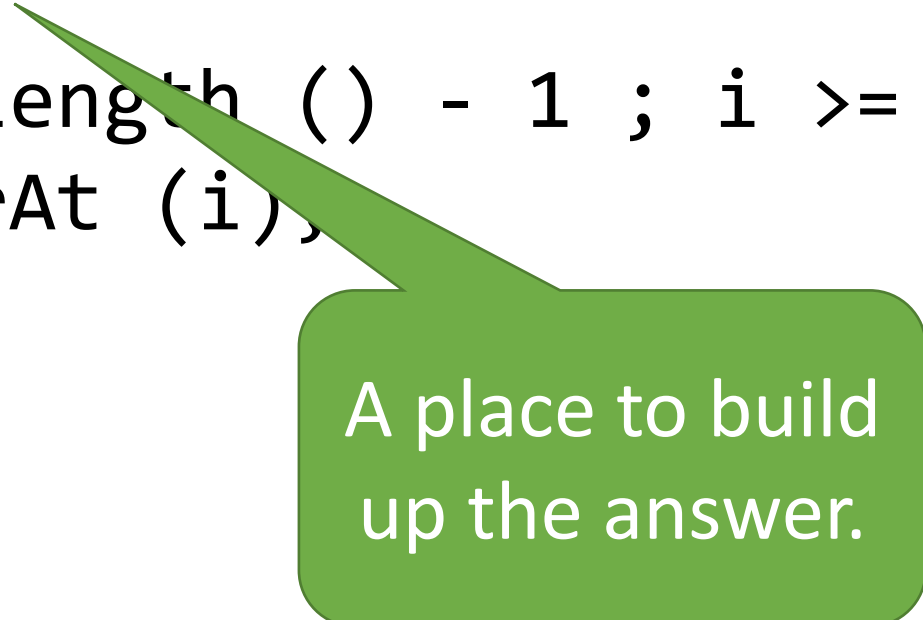
```
public String reverse (String s){  
    String ans = "";  
    for (int i = s.length () - 1 ; i >= 0 ; i--)  
        ans += s.charAt (i);  
    return ans;  
}  
}
```

```
public String reverse (String s) {  
    String ans = "";  
  
    for (int i = s.length () - 1 ; i >= 0 ; i--)  
        ans += s.charAt (i);  
  
    return ans;  
}
```

0	1	2	3	4	5	6	7	8	9	10	11	12
V	i	o	l	a		D	e	s	m	o	n	d



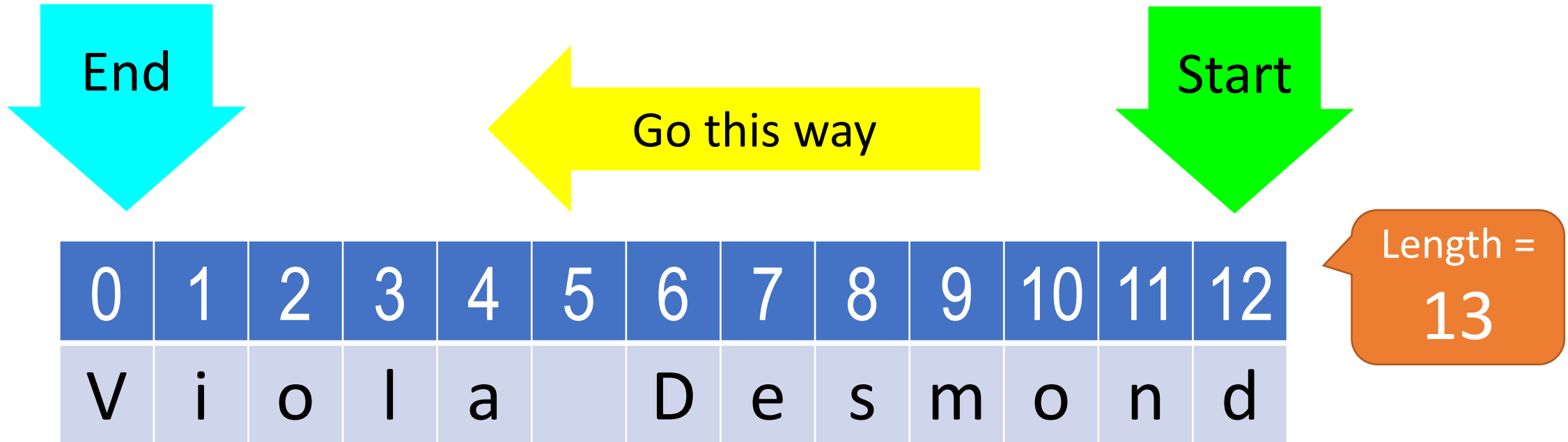
```
public String reverse (String s) {  
    String ans = "";  
    for (int i = s.length () - 1 ; i >= 0 ; i--)  
        ans += s.charAt (i);  
    return ans;  
}
```



A place to build up the answer.

0	1	2	3	4	5	6	7	8	9	10	11	12
V	i	o	l	a		D	e	s	m	o	n	d

```
public String reverse (String s) {  
    String ans = "";  
    for (int i = s.length () - 1 ; i >= 0 ; i--)  
        ans += s.charAt (i);  
    return ans;  
}
```



```

public String reverse (String s) {
    String ans = "";

    for (int i = s.length () - 1 ; i >= 0 ; i--)
        ans += s.charAt (i);
    return ans;
}

```

ans values

d
 dn
 dno
 dnom
 dnoms
 dnomse

dnomseD
 dnomseD (space)
 dnomseD a
 dnomseD al
 dnomseD ali
 dnomseD alio
 dnomseD alioV

0	1	2	3	4	5	6	7	8	9	10	11	12
V	i	o	l	a		D	e	s	m	o	n	d



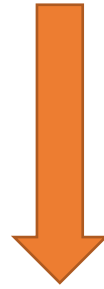
Caesar
Shift



Alice

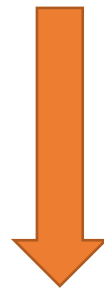


Plaintext



Key: shift of 1 letter

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
 Z A B C D E F G H I J K L M N O P Q R S T U V W X Y



Ciphertext



Bob



Alice

HELP

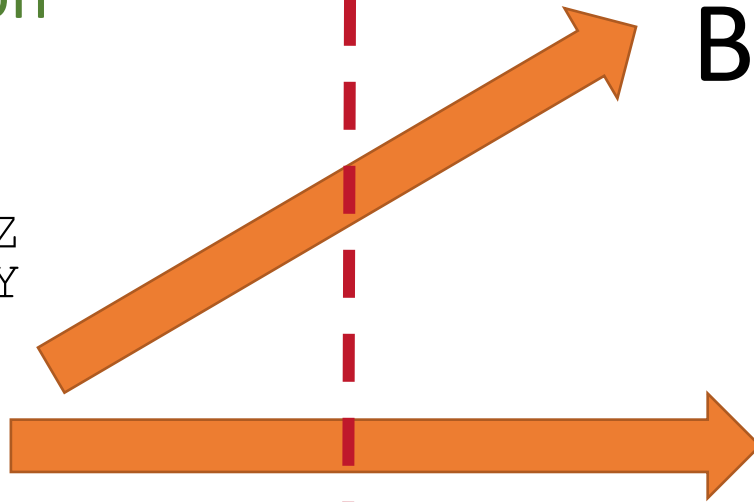


Encryption

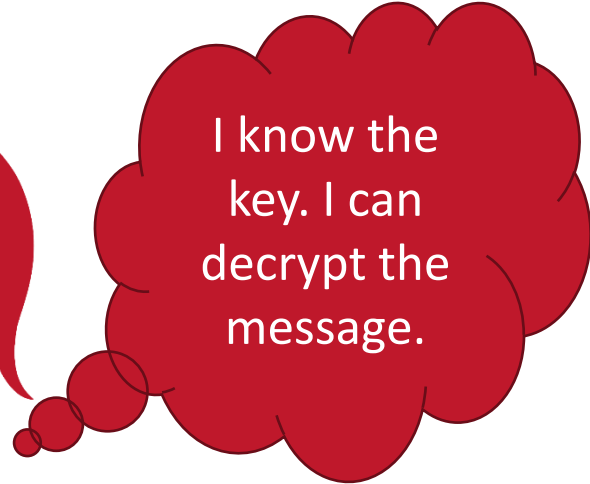
Key: shift of 1 letter

ABCDEFGHIJKLMNOPQRSTUVWXYZ
ZABCDEFGHIJKLMNOPQRSTUVWXYZ

GDKO



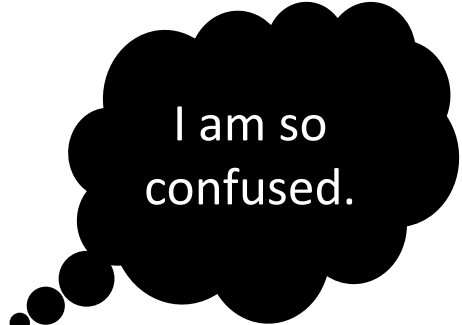
Bob



I know the key. I can decrypt the message.



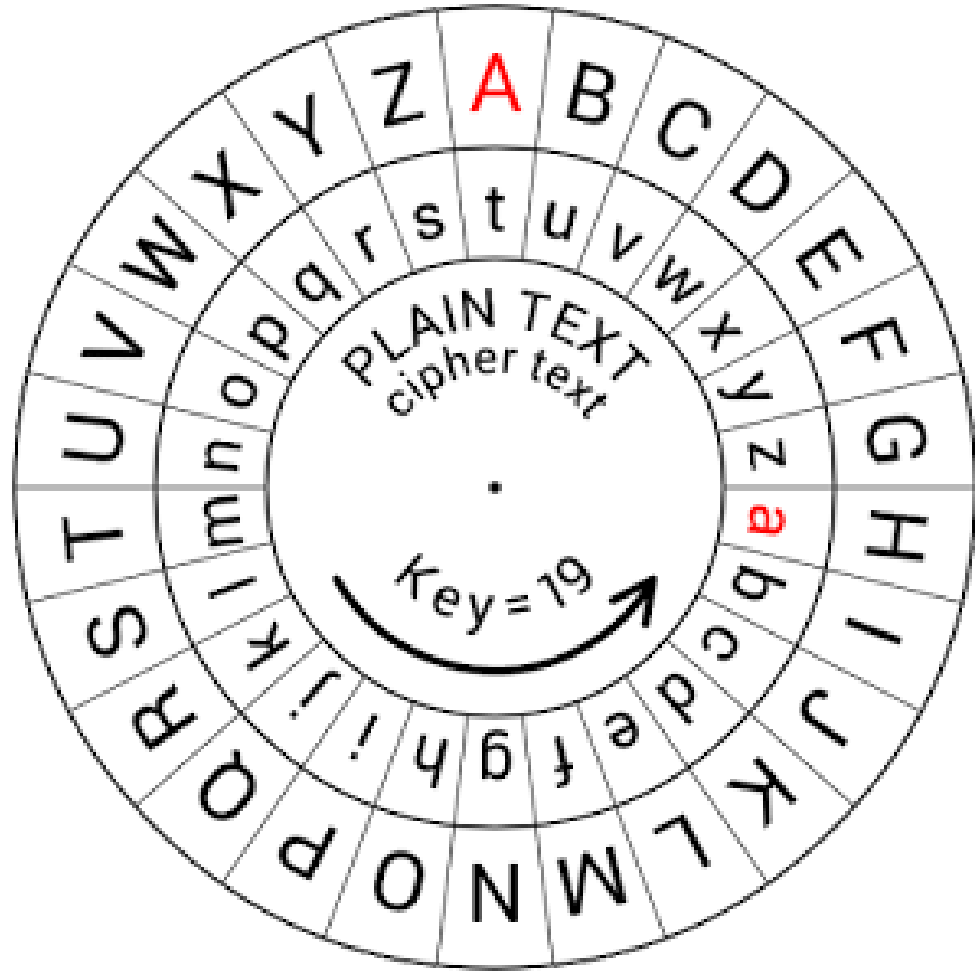
Eve



I am so confused.



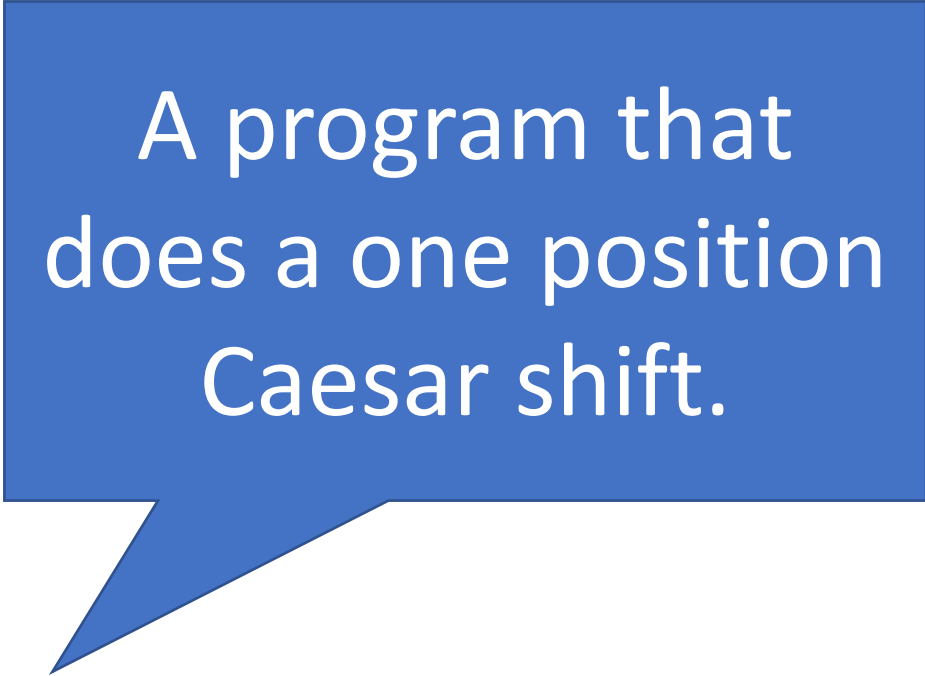
Yeah. Not really.



```
public class encrypt {
    public static void main (String args[]) {
        new encrypt ();
    }

    public encrypt () {
        String s = "babyz";
        System.out.println (caesar (s));
    }

    public String caesar (String s) {
        String ans = "";
        s = s.toLowerCase ();
        for (int i = 0 ; i < s.length () ; i++) {
            int ascii = (int) (s.charAt (i));
            if (ascii == 32) //space
                ans += ' ';
            else if (ascii == 122) //z
                ans += 'a';
            else //all other letters
                ans += (char) (ascii + 1);
        }
        return ans;
    }
}
```



A program that
does a one position
Caesar shift.

```
public String caesar (String s) {  
    String ans = "";  
    s = s.toLowerCase ();  
    for (int i = 0 ; i < s.length () ; i++) {  
        int ascii = (int) (s.charAt (i));  
        if (ascii == 32) //space  
            ans += ' ';  
        else if (ascii == 122) //z  
            ans += 'a';  
        else //all other letters  
            ans += (char) (ascii + 1);  
    }  
    return ans;  
}
```

Starts and
ends where?

What does
this do?

```
public String caesar (String s) {  
    String ans = "";  
    s = s.toLowerCase ();  
    for (int i = 0 ; i < s.length () ; i++) {  
        int ascii = (int) (s.charAt (i));  
        if (ascii == 32) //space  
            ans += ' ';  
        else if (ascii == 122) //z  
            ans += 'a';  
        else //all other letters  
            ans += (char) (ascii + 1);  
    }  
    return ans;  
}
```

Take the
next
letter.

What does
this do?

```
public String caesar (String s) {  
    String ans = "";  
    s = s.toLowerCase ();  
    for (int i = 0 ; i < s.length () ; i++) {  
        int ascii = (int) (s.charAt (i));  
        if (ascii == 32) //space  
            ans += ' '  
        else if (ascii == 122) //z  
            ans += 'a';  
        else //all other letters  
            ans += (char) (ascii + 1);  
    }  
    return ans;  
}
```

Add on the
shifted next
letter

What does this do?

The image shows the interior of a large, domed structure, likely a historical building or a modern architectural model. The ceiling is a dome covered in a repeating pattern of circular, sunburst-like motifs. Below the dome, there are three arched windows that offer a view of a bright blue sky with scattered white clouds. The walls and floor are made of a light-colored, textured material, possibly stone or concrete, and show signs of wear and age. The overall atmosphere is one of historical grandeur and architectural detail.

Another method


```
public String caesar2 (String s) {  
    String alpha = "abcdefghijklmnopqrstuvwxyz ";  
    String key = "bcdefghijklmnopqrstuvwxyz a";  
    String ans = "";  
    s = s.toLowerCase ();  
    for (int i = 0 ; i < s.length () ; i++) {  
        char letter = s.charAt (i);  
        int place = alpha.indexOf (letter);  
        ans += key.charAt (place);  
    }  
    return ans;  
}
```

0	1	2	3	4
V	i	o	l	a

```
public String caesar2 (String s) {
    String alpha = "abcdefghijklmnopqrstuvwxyz ";
    String key = "bcdefghijklmonpqrstuvwxyza ";
    String ans = "";
    s = s.toLowerCase ();
    for (int i = 0 ; i < s.length () ; i++) {
        char letter = s.charAt (i);
        int place = alpha.indexOf (letter);
        ans += key.charAt (place);
    }
    return ans;
}
```

0	1	2	3	4
v	i	o	l	a

These three lines can all be done in one glorious line of code. However, it tends to freak students out, so I trace the first one.

```
for (int i = 0 ; i < s.length () ; i++) {  
    char letter = s.charAt (i);  
    int place = alpha.indexOf (letter);  
    ans += key.charAt (place);  
}
```

```
public String caesar2 (String s) {  
    String alpha = "abcdefghijklmnopqrstuvwxyz ";  
    String key = "bcdefghijklmonpqrstuvwxyz a ";  
    String ans = "";  
    s = s.toLowerCase ();  
    for (int i = 0 ; i < s.length () ; i++)  
        ans += key.charAt (alpha.indexOf (s.charAt (i)));  
    return ans;  
}
```