JAVA WORDS

void

int **double**

String

char

boolean

for

while **true** **false**

if **else**

return

VARIABLES (Post-It Notes)

Create

```
int  sum  =  0;
```

Replace

```
sum  =  15;
```

Update

```
sum  =  sum  +  15;
```

DECISIONS

```
if ( some-condition )
{
    things to do when
    condition is true
}
```

```
if ( some-condition )
{
    things to do when
    condition is true
}
else
{
    things to do when
    condition is false
       OPTIONAL
}
```

if-else-if vs. if-if-if

```
if ( some-condition )
{

}
else if ( some-condition )
{

}
else if ( some-condition )
{

}
```

- *first condition to succeed breaks the chain*
- *at most one will execute*

```
if ( some-condition )
{

}
if ( some-condition )
{

}
if ( some-condition )
{

}
```

- *independent if's*
- *all might execute*

while vs. for

```
int count = 0;

while ( count < n )
{
    ...
    Do the important work
    ...

    count = count + 1;
}
```

```
for ( int count = 0 ; count < n ; count = count + 1 )
{
    ...
    Do the important work
    ...
}
```

GENERAL PATTERN

```
SET UP counter, result, other Post-Its

while ( there is work to do )
{
    ...
    Do one piece of work
    ...

    Update Post-Its
}
```

boolean methods

```
if ( isVowel( letter ) == true )
{
    ...
    code for vowel
    ...
}
else
{
    ...
    code for NOT vowel
    ...
}
```

```
if ( isVowel( letter ) )
{
    ...
    code for vowel
    ...
}
else
{
    ...
    code for NOT vowel
    ...
}
```

boolean methods

```
if ( isVowel( letter ) == false )
{
    ...
    code for NOT vowel
    ...
}
else
{
    ...
    code for vowel
    ...
}
```

```
if ( !isVowel( letter ) )
{
    ...
    code for NOT vowel
    ...
}
else
{
    ...
    code for vowel
    ...
}
```

boolean methods

```
if ( (x < y) )
{
    ...
    code for x < y
    ...
}
else
{
    ...
    code for x ≮ y, ie x ≥ y
    ...
}
```

```
if ( !(x < y) )
{
    ...
    code for x ≮ y, ie x ≥ y
    ...
}
else
{
    ...
    code for x < y
    ...
}
```

boolean methods

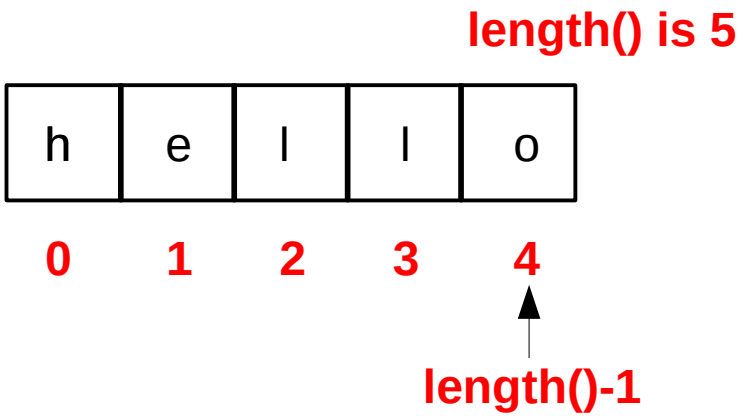
```
if ( isVowel( letter ) == false )
{
    ...
    code for NOT vowel
    ...
}
else
{
    ...
    code for vowel
    ...
}
```

```
if ( !isVowel( letter ) )
{
    ...
    code for NOT vowel
    ...
}
else
{
    ...
    code for vowel
    ...
}
```

Strings and chars

Create

```
String word = “hello”;
```



Access

```
word.charAt( some-index )

word.charAt( 0 )

word.charAt( 1 )

word.charAt( 4 )

word.charAt( 5 )   error, no such cell
```

Length

```
word.length()
```

GENERAL PATTERN

```
SET UP result, other Post-Its

// In English: for each index in the phrase
for ( int index = 0; index < phrase.length(); index = index + 1 )
{
    ...
    Extract letter

    Do s.th. with letter if necessary
    ...
}
```

Complicated Conditions

(&& ||)

&& vs ||

condition1 && condition2

both (all) conditions must be true
for the whole thing to be true

condition1 || condition2

at least one conditions must be true
for the whole thing to be true

&& vs ||

(x >= 0) && (x <= 800)

both (all) must be true
for the point to be **inside** the canvas

(x < 0) || (x > 800)

at least one must be true
for the point to be **outside** the canvas

&& vs ||

condition1 && condition2

condition1	condition2	cond1&&cond2
T	**T**	**T**
T	**F**	**F**
F	**T**	**F**
F	**F**	**F**

Both must be True

condition1 || condition2

condition1	condition2	cond1||cond2
T	**T**	**T**
T	**F**	**T**
F	**T**	**T**
F	**F**	**F**

At least one must be True

&& vs ||

&& has higher pecedence than ||

(just like * has higher precedence than +)

((x > 5) || (x < 8)) && (y > 3) (2 + 3) * 4

different from different from

(x > 5) || ((x < 8) && (y > 3)) 2 + (3 * 4)

if the ( ) are omitted

(x > 5) || (x < 8) && (y > 3) 2 + 3 * 4

interpreted the same as same as

(x > 5) || ((x < 8) && (y > 3)) 2 + (3 * 4)