

# More practice with Regular Expressions



For String manipulation



## By the end of this video you will be able to...

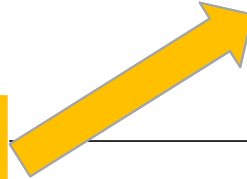
- Write the regular expressions you will need to do the programming assignment in this module

# Relating regex's to the project

```
public abstract class Document
{
    // The text of the whole document
    private String text;

    protected List<String> getTokens(String pattern)
    {
```

**given helper method**



# Relating regex's to the project

```
public abstract class Document
{
    // The returns a List of "tokens" document regular expression defining the "tokens"
    private ...
    protected List<String> getTokens(String pattern)
    {
        ...
    }
}
```

```
public abstract class Document {  
    private String text;  
    protected List<String> getTokens(String pattern)  
    { ... }  
    ...  
}
```

```
public class BasicDocument extends Document  
{  
    ...  
    public int getNumWords() {  
        List<String> tokens = getTokens( ??? );  
        return tokens.size();  
    }  
}
```

**Need a regex that  
matches "any word"**

```
public class BasicDocument extends Document
{
    ...
    public int getNumWords() {
        List<String> tokens = getTokens( ??? );
        return tokens.size();
    }
}
```

**What constitutes a word?**


**"Any contiguous sequence of alphabetic characters"**

```
public class BasicDocument extends Document
{
    ...
    public int getNumWords() {
        List<String> tokens = getTokens( "[a-zA-Z]" );
        return tokens.size();
    }
}
```

**What constitutes a word?**

**"Any contiguous sequence of alphabetic characters"**

```
public class BasicDocument extends Document
{
    ...
    public int getNumWords() {
        List<String> tokens = getTokens( "[a-zA-Z]" );
        return tokens.size();
    }
}
```




**What constitutes a word?**

**"Any contiguous sequence of alphabetic characters"**



```
public class BasicDocument extends Document
{
    ...
    public int getNumWords() {
        List<String> tokens = getTokens( "[a-zA-Z]+" );
        return tokens.size();
    }
}
```



**What constitutes a word?**

**"Any contiguous sequence of alphabetic characters"**

```
public class BasicDocument extends Document
{
    ...
    public int getNumWords() {
        List<String> tokens = getTokens( "[a-zA-Z]+" );
        return tokens.size();
    }
}
```

**What constitutes a word?**

**"Any contiguous sequence of alphabetic characters"**

```
public class BasicDocument extends Document
{
    ...
    public int getNumSentences() {
        List<String> tokens = getTokens( ??? );
        return tokens.size();
    }
}
```

**What constitutes a sentence?**

**"A sequence of any characters ending with  
end of sentence punctuation (. ! ?)"**

```
public class BasicDocument extends Document
{
    ...
    public int getNumSentences() {
        List<String> tokens = getTokens( ??? );
        return tokens.size();
    }
}
```

**What constitutes a sentence?**


**"A contiguous sequence of characters that does NOT include end of sentence punctuation."**

```
public class BasicDocument extends Document
{
    ...
    public int getNumSentences() {
        List<String> tokens = getTokens( "[^!?.]+" );
        return tokens.size();
    }
}
```

**What constitutes a sentence?**

**"A contiguous sequence of characters that does NOT include end of sentence punctuation."**

```
public class BasicDocument extends Document
{
    ...
    public int getNumSentences() {
        List<String> tokens = getTokens( "[^!?.]+" );
        return tokens.size();
    }
}
```



**What constitutes a sentence?**

**"A contiguous sequence of characters that does NOT include end of sentence punctuation."**

```
public class BasicDocument extends Document
{
    ...
    public int getNumSentences() {
        List<String> tokens = getTokens( "[^!?.]+" );
        return tokens.size();
    }
}
```

**What constitutes a sentence?**

**"A contiguous sequence of characters that does NOT include end of sentence punctuation."**