

Identifying Genes

Using While Loops

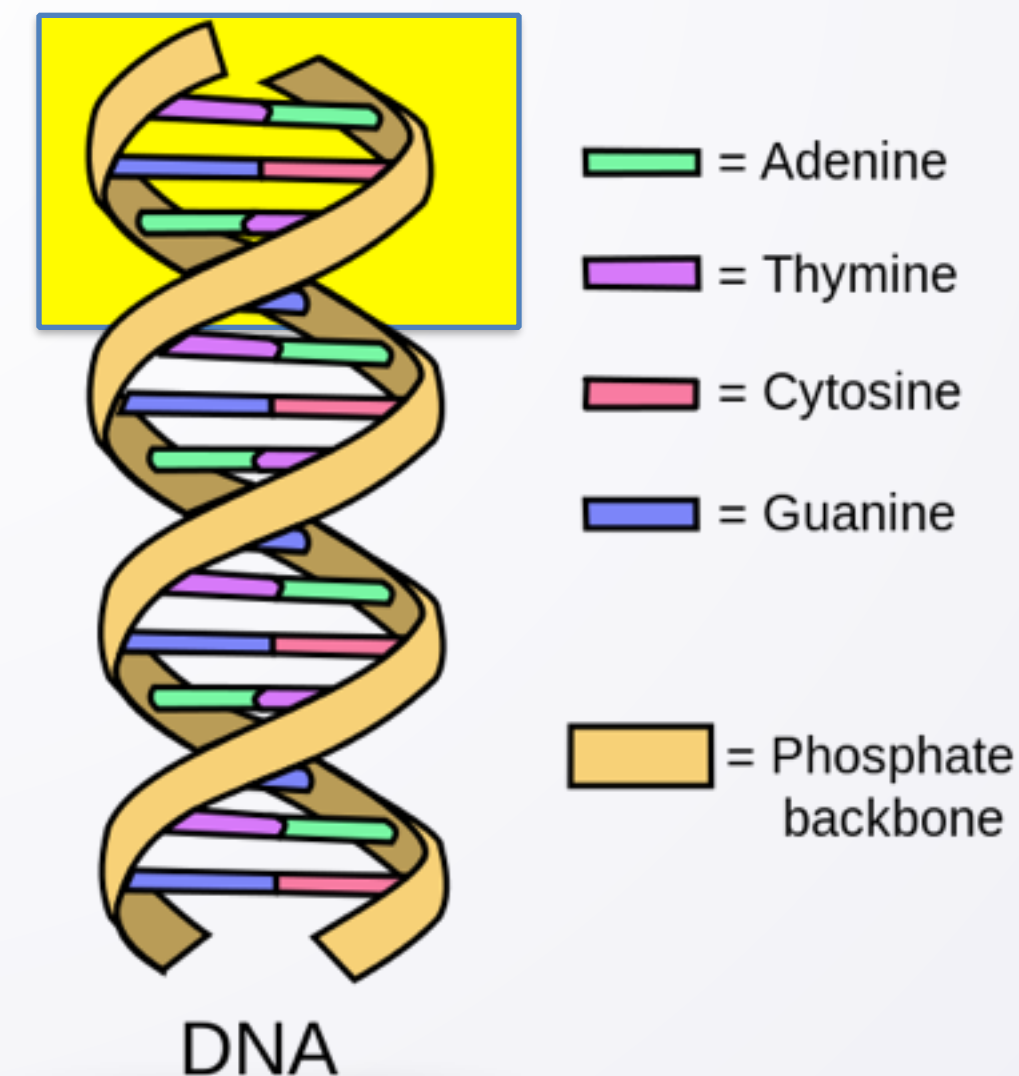
Searching for Many Genes

- New programming tool: **while loop**
 - Loop over code-block until time to stop
 - Different from for loop with iterables



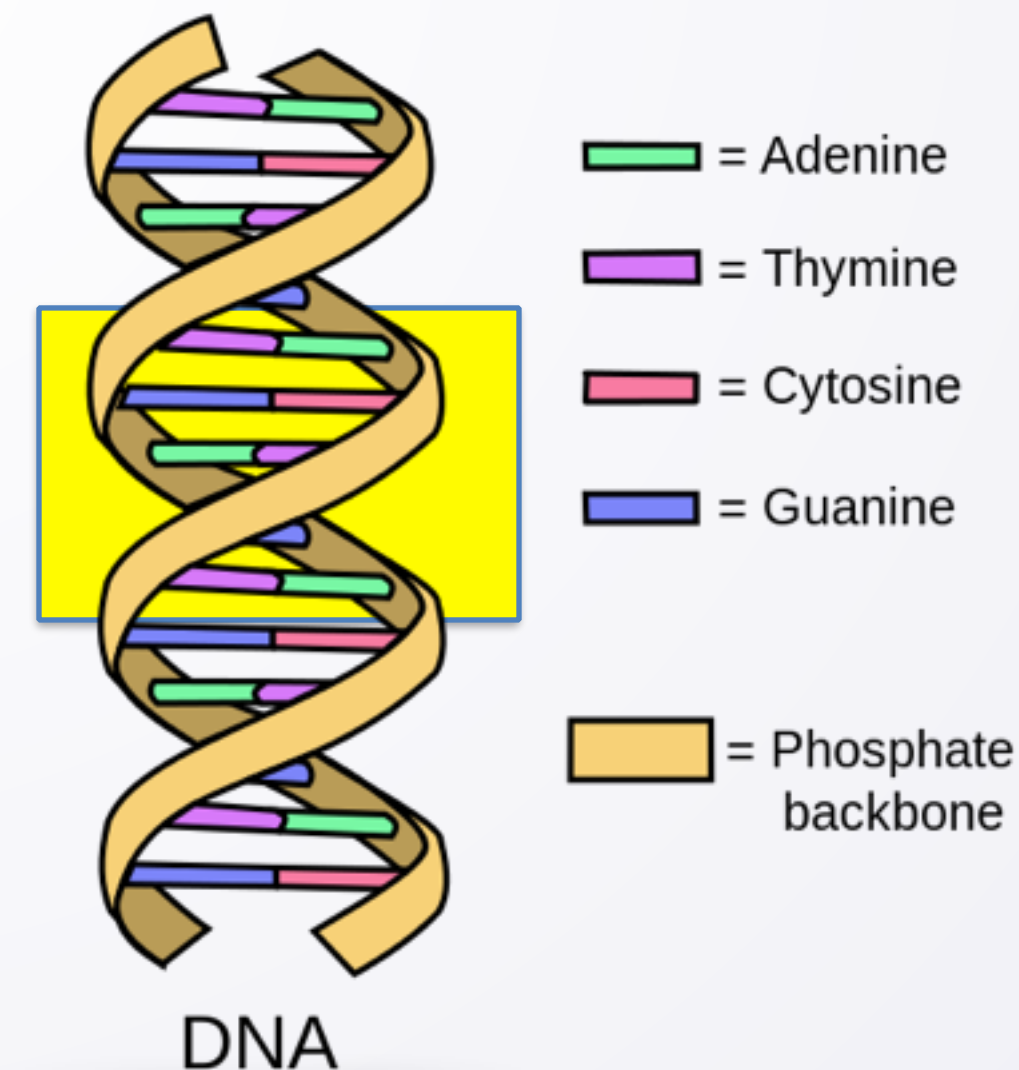
Searching for Many Genes

- New programming tool: **while loop**
 - Loop over code-block until time to stop
 - Different from for loop with iterables
- To find all genes in a strand of DNA:
 - Find start codon: "ATG"



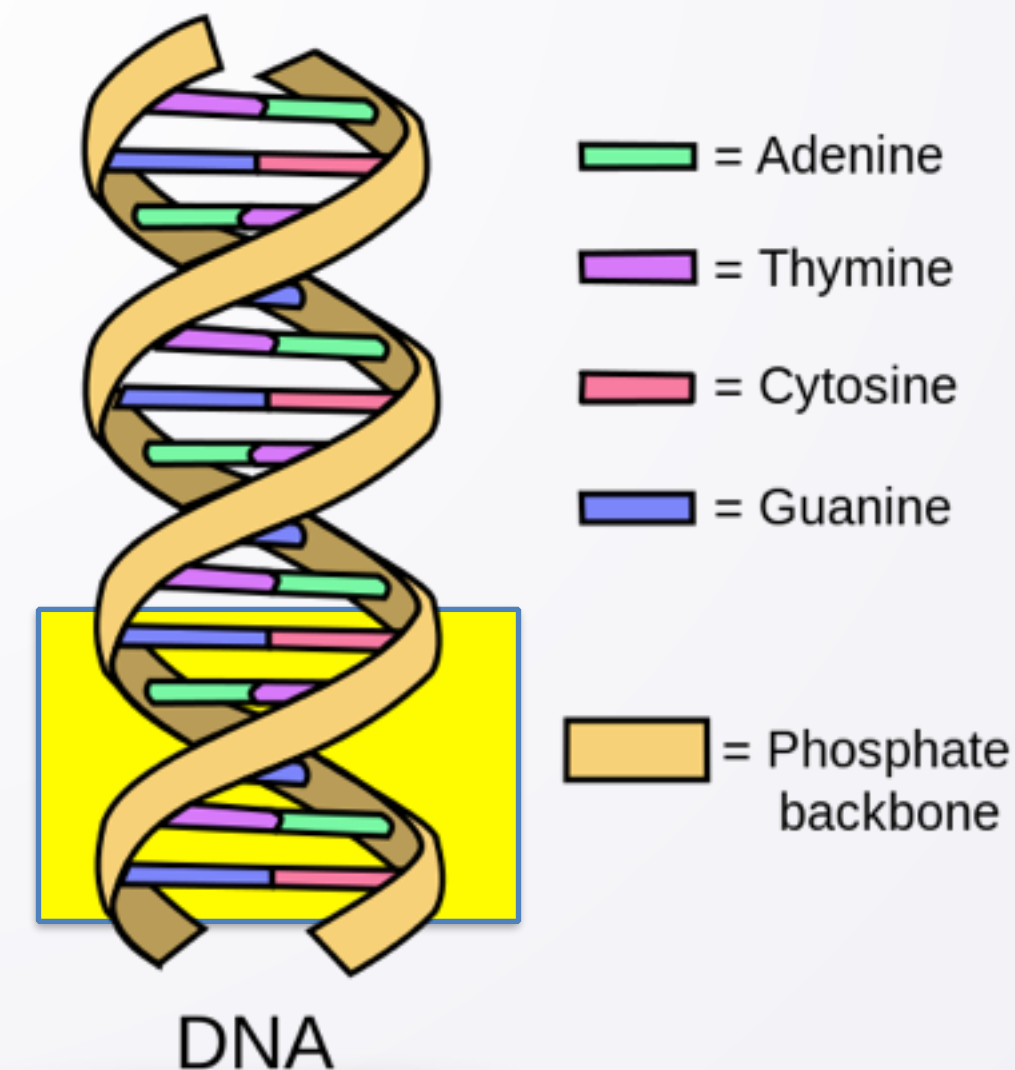
Searching for Many Genes

- New programming tool: **while loop**
 - Loop over code-block until time to stop
 - Different from for loop with iterables
- To find all genes in a strand of DNA:
 - Find start codon: "ATG"
 - Find another one further along



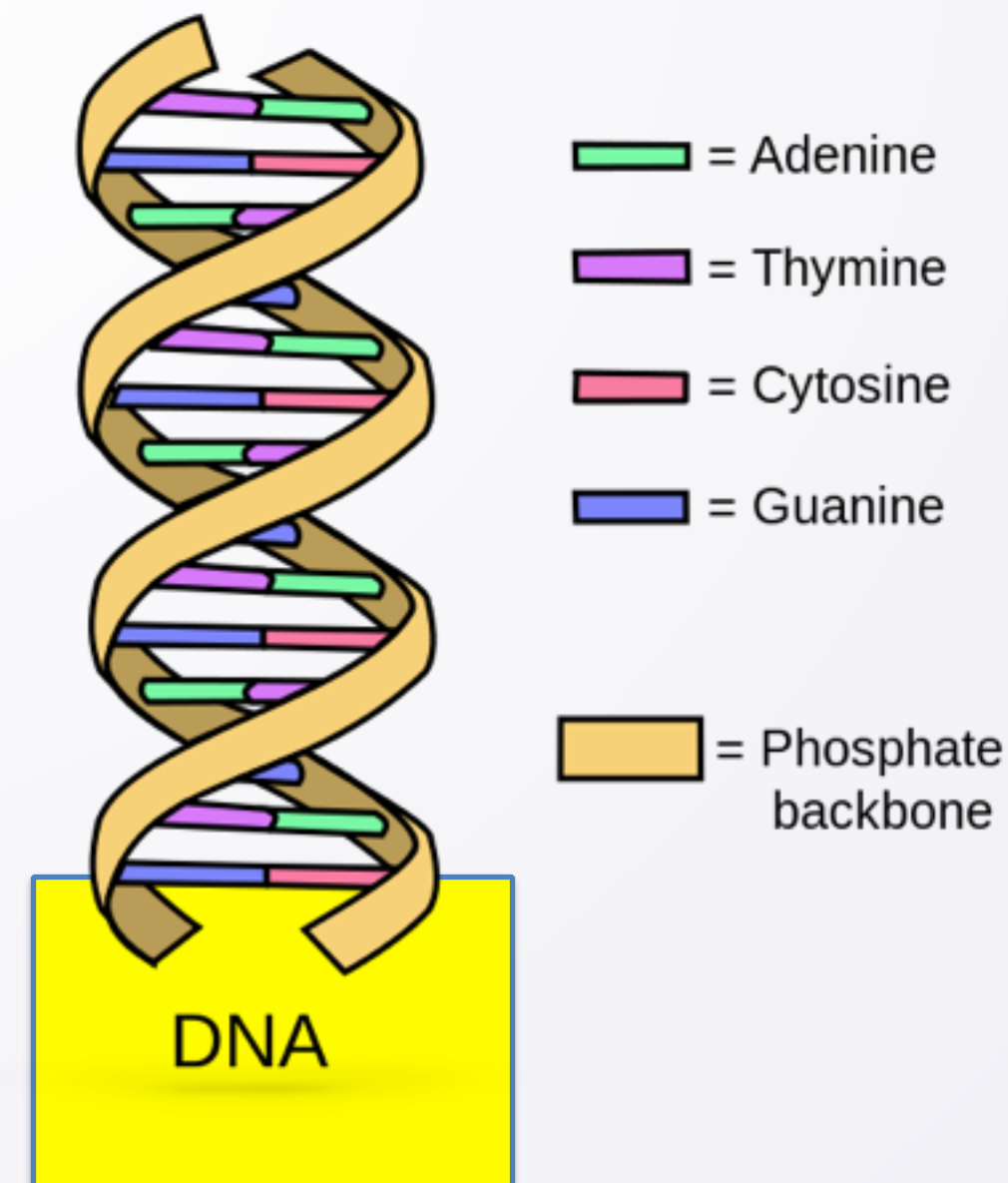
Searching for Many Genes

- New programming tool: **while loop**
 - Loop over code-block until time to stop
 - Different from for loop with iterables
- To find all genes in a strand of DNA:
 - Find start codon: "ATG"
 - Find another one further along
 - Find another one further along



Searching for Many Genes

- New programming tool: **while loop**
 - Loop over code-block until time to stop
 - Different from for loop with iterables
- To find all genes in a strand of DNA:
 - Find start codon: "ATG"
 - Find another one further along
 - Find another one further along
 - When to stop searching ?



From Finding One to Finding All

- Start with code we know is correct

```
public String findProtein(String dna){  
    int start = dna.indexOf("atg");  
    if (start == -1) {  
        return "";  
    }  
    int stop = dna.indexOf("tag", start+3);  
    if ((stop - start) % 3 == 0){  
        return dna.substring(start, stop+3);  
    }  
    else {  
        return "";  
    }  
}
```

From Finding One to Finding All

- Start with code we know is correct
 - Change to use while loop: find all start codons

```
int start = dna.indexOf("atg");  
if (start == -1) {  
    return "";  
}  
  
if ((stop - start) % 3 == 0){  
    return dna.substring(start, stop+3);  
}  
else {  
    return "";  
}  
}
```


Finding All Start Codons

- Code to indicate no start codon found

```
int start = dna.indexOf("atg");  
if (start == -1) {  
    return "";  
}
```

Finding All Start Codons

- Code to indicate no start codon found
 - Use this idea to know when to stop looking
 - Use two-parameter **indexOf** as part of looping

```
public void printAllStarts(String dna) {  
    int start = 0;  
    while (true){  
        int loc = dna.indexOf("atg", start);  
        if (loc == -1) {  
            break;  
        }  
        System.out.println("starts at "+loc);  
        start = loc + 3;  
    }  
}
```

Finding All Start Codons

- Code to indicate no start codon found
 - Use this idea to know when to stop looking
 - Use two-parameter **indexOf** as part of looping
 - Be able to exit loop at some point

```
public void printAllStarts(String dna) {  
    int start = 0;  
    while (start < dna.length()) {  
        int loc = dna.indexOf("ATG", start);  
        if (loc == -1) {  
            break;  
        }  
        printStart(dna, loc);  
        start = loc + 3;  
    }  
}
```

Understanding Loops

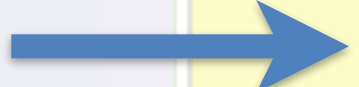
```
String dna = "cccatgcccatgcccatgccccatgcccatgatg";  
              01234567890123456789012345678901234
```

```
public void printAllStarts(String dna) {  
    int start = 0;  
    while (true){  
        int loc = dna.indexOf("atg",start);  
        if (loc == -1) {  
            break;  
        }  
        System.out.println("starts at "+loc);  
        start = loc + 3;  
    }  
}
```


Understanding Loops

start = 0 loc = ?

String dna = "cccatgcccatgcccatgccccatgcccatgatg";
01234567890123456789012345678901234

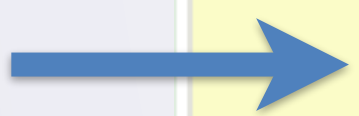


```
public void printAllStarts(String dna) {  
    int start = 0;  
    while (true){  
        int loc = dna.indexOf("atg",start);  
        if (loc == -1) {  
            break;  
        }  
        System.out.println("starts at "+loc);  
        start = loc + 3;  
    }  
}
```

Understanding Loops

start = 0 loc = 3

String dna = "cccatgccccatgccccatgccccatgatg";
01234567890123456789012345678901234

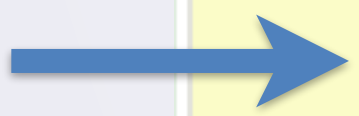


```
public void printAllStarts(String dna) {  
    int start = 0;  
    while (true){  
        int loc = dna.indexOf("atg",start);  
        if (loc == -1) {  
            break;  
        }  
        System.out.println("starts at "+loc);  
        start = loc + 3;  
    }  
}
```

Understanding Loops

start = 0 loc = 3

String dna = "cccatgccccatgccccatgccccatgccccatgatg";
01234567890123456789012345678901234



```
public void printAllStarts(String dna) {  
    int start = 0;  
    while (true){  
        int loc = dna.indexOf("atg",start);  
        if (loc == -1) {  
            break;  
        }  
        System.out.println("starts at "+loc);  
        start = loc + 3;  
    }  
}
```



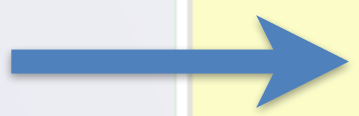
starts at 3

Understanding Loops

start = 6 loc = 3

String dna = "cccatgccccatgccccatgccccatgccccatgatg";
01234567890123456789012345678901234

```
public void printAllStarts(String dna) {  
    int start = 0;  
    while (true){  
        int loc = dna.indexOf("atg",start);  
        if (loc == -1) {  
            break;  
        }  
        System.out.println("starts at "+loc);  
        start = loc + 3;  
    }  
}
```




starts at 3

Understanding Loops

start = 6 loc = 9

String dna = "cccatgccccatgccccatgccccatgccccatgatg";
01234567890123456789012345678901234



```
public void printAllStarts(String dna) {  
    int start = 0;  
    while (true){  
        int loc = dna.indexOf("atg",start);  
        if (loc == -1) {  
            break;  
        }  
        System.out.println("starts at "+loc);  
        start = loc + 3;  
    }  
}
```




starts at 3

Understanding Loops

start = 6 loc = 9

String dna = "cccatgccc**atg**cccatgccccatgccccatgatg";
01234567890123456789012345678901234

```
public void printAllStarts(String dna) {  
    int start = 0;  
    while (true){  
        int loc = dna.indexOf("atg",start);  
        if (loc == -1) {  
            break;  
        }  
        System.out.println("starts at "+loc);  
        start = loc + 3;  
    }  
}
```




starts at 3
starts at 9


Understanding Loops

start = 6 loc = 9

String dna = "cccatgcccatgcccatgccccatgcccatgatg";
01234567890123456789012345678901234



```
public void printAllStarts(String dna) {  
    int start = 0;  
    while (true){  
        int loc = dna.indexOf("atg",start);  
        if (loc == -1) {  
            break;  
        }  
        System.out.println("starts at "+loc);  
        start = loc + 3;  
    }  
}
```

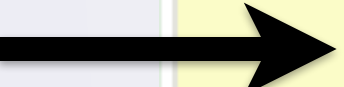


```
starts at 3  
starts at 9  
starts at 15  
starts at 22  
starts at 29  
starts at 32
```


Understanding Loops

start = 35 loc = 32

String dna = "cccatgcccatgcccatgccccatgcccatgatg";
01234567890123456789012345678901234



```
public void printAllStarts(String dna) {  
    int start = 0;  
    while (true){  
        int loc = dna.indexOf("atg",start);  
        if (loc == -1) {  
            break;  
        }  
        System.out.println("starts at "+loc);  
        start = loc + 3;  
    }  
}
```




```
starts at 3  
starts at 9  
starts at 15  
starts at 22  
starts at 29  
starts at 32
```



Understanding Loops

start = 35 loc = -1

String dna = "cccatgcccatgcccatgccccatgcccatgatg";
01234567890123456789012345678901234



```
public void printAllStarts(String dna) {  
    int start = 0;  
    while (true){  
        int loc = dna.indexOf("atg",start);  
        if (loc == -1) {  
            break;  
        }  
        System.out.println("starts at "+loc);  
        start = loc + 3;  
    }  
}
```



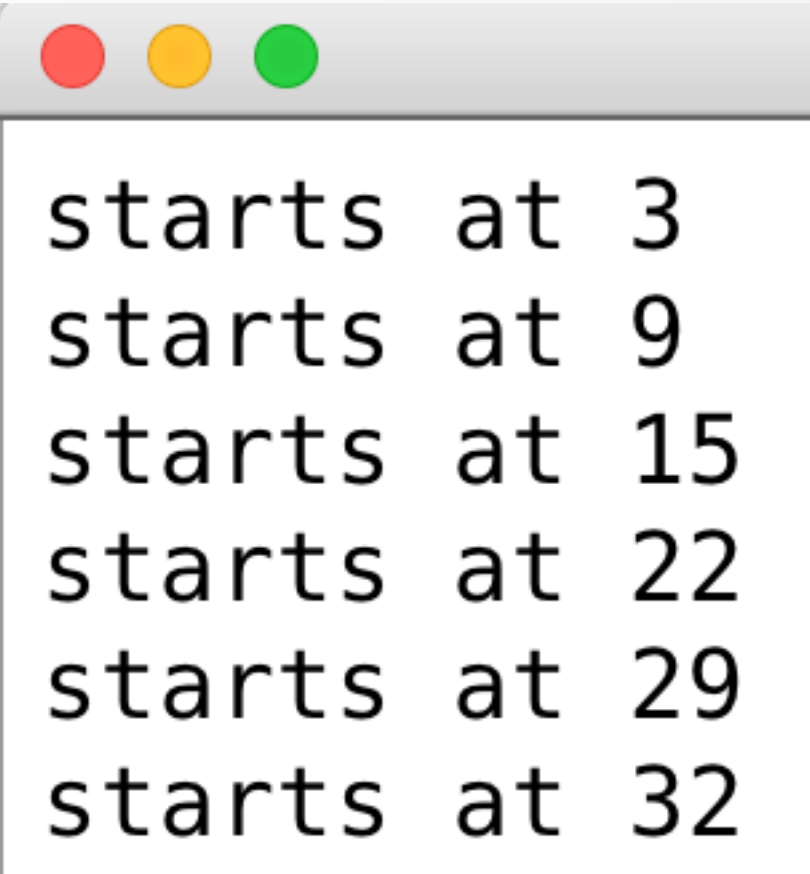
```
starts at 3  
starts at 9  
starts at 15  
starts at 22  
starts at 29  
starts at 32
```

Understanding Loops

start = 35 loc = -1

String dna = "cccatgcccatgcccatgccccatgcccatgatg";
01234567890123456789012345678901234

```
public void printAllStarts(String dna) {  
    int start = 0;  
    while (true){  
        int loc = dna.indexOf("atg",start);  
        if (loc == -1) {  
            break;  
        }  
        System.out.println("starts at "+loc);  
        start = loc + 3;  
    }  
}
```



```
starts at 3  
starts at 9  
starts at 15  
starts at 22  
starts at 29  
starts at 32
```

From Start to Stop: Finding Genes

- We want to find all genes
 - Walked through loop for finding start "ATG"
 - Need to find stop: "TAG", "TAA", "TGA"

```
public void printAllStarts(String dna) {  
    int start = 0;  
    while (true){  
        int loc = dna.indexOf("atg",start);  
        if (loc == -1) {  
            break;  
        }  
        System.out.println("starts at "+loc);  
        start = loc + 3;  
    }  
}
```

From Start to Stop: Finding Genes

- We want to find all genes
 - Walked through loop for finding start "ATG"
 - Need to find stop: "TAG", "TAA", "TGA"

| | | | | | | | | | | | | | | | | | | | | | |
|-----|-----|-----|---|-----|-------|-----|--------|-----|----|---|---|---|---|---|---|---|---|---|---|---|---|
| CCC | ATG | xxx | y | TAG | xxxxx | TGA | yyyyyy | TAA | xx | | | | | | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |

- Find each stop codon (after start)
 - If found, and multiple of 3 from start, OK

From Start to Stop: Finding Genes

- We want to find all genes
 - Walked through loop for finding start "ATG"
 - Need to find stop: "TAG", "TAA", "TGA"

| | | | | | | | | | | | | | | | | | | | | | |
|-----|-----|-----|---|-----|-------|-----|--------|-----|----|---|---|---|---|---|---|---|---|---|---|---|---|
| CCC | ATG | xxx | y | TAG | xxxxx | TGA | yyyyyy | TAA | xx | | | | | | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |

- Find each stop codon (after start)
 - If found, and multiple of 3 from start, OK
 - Choose closest of those that are OK

From Start to Stop: Finding Genes

- We want to find all genes
 - Walked through loop for finding start "ATG"
 - Need to find stop: "TAG", "TAA", "TGA"



- Find each stop codon (after start)
 - If found, and multiple of 3 from start, OK
 - Choose closest of those that are OK

Translating Algorithm to Code

- Find each stop codon (after start)
 - If found, and multiple of 3 from start, OK
 - Choose closest of those that are OK

```
/**
 * Find a valid stop codon in dna that occurs after index.
 * If no valid stop codon found, return dna.length()
 * @param dna is String being searched
 * @param index is index where search starts
 * @return index of beginning of a valid stop codon,
 * or dna.length() if no valid codon
 */
public int findStopIndex(String dna, int index)
```

Translating Algorithm to Code

```
public int findStopIndex(String dna, int index){  
    int stop1 = dna.indexOf("tga", index);  
    if (stop1 == -1 || (stop1-index) % 3 != 0){  
        stop1 = dna.length();  
    }  
    int stop2 = dna.indexOf("taa", index);  
    if (stop2 == -1 || (stop2-index) % 3 != 0){  
        stop2 = dna.length();  
    }  
    int stop3 = dna.indexOf("tag", index);  
    if (stop3 == -1 || (stop3-index) % 3 != 0){  
        stop3 = dna.length();  
    }  
    return Math.min(stop1, Math.min(stop2, stop3));  
}
```


Translating Algorithm to Code

```
public int findStopIndex(String dna, int index){  
    int stop1 = dna.indexOf("tga", index);  
    if (stop1 == -1 || (stop1-index) % 3 != 0){  
        stop1 = dna.length();  
    }  
    int stop2 = dna.indexOf("taa", index);
```

- Short-circuit boolean evaluation
 - if stop1 equals -1 ? No need to continue
 - For A || B, if A is true, don't evaluate B
 - For A && B, if A is false, don't evaluate B

```
        stop3 = dna.length();  
    }  
    return Math.min(stop1, Math.min(stop2, stop3));  
}
```

Translating Algorithm to Code

```
public int findStopIndex(String dna, int index){  
    int stop1 = dna.indexOf("tga", index);  
    if (stop1 == -1 || (stop1-index) % 3 != 0){  
        stop1 = dna.length();  
    }  
    int stop2 = dna.indexOf("taa", index);  
    if (stop2 == -1 || (stop2-index) % 3 != 0){  
        stop2 = dna.length();  
    }  
    int stop3 = dna.indexOf("tag", index);  
    if (stop3 == -1 || (stop3-index) % 3 != 0){  
        stop3 = dna.length();  
    }  
    return Math.min(stop1, Math.min(stop2, stop3));  
}
```

Translating Algorithm to Code

```
public int findStopIndex(String dna, int index){  
    int stop1 = dna.indexOf("tga", index);  
    if (stop1 == -1 || (stop1-index) % 3 != 0){  
        stop1 = dna.length();  
    }  
    int stop2 = dna.indexOf("taa", index);  
    if (stop2 == -1 || (stop2-index) % 3 != 0){  
        stop2 = dna.length();  
    }  
    int stop3 = dna.indexOf("tag", index);  
    if (stop3 == -1 || (stop3-index) % 3 != 0){  
        stop3 = dna.length();  
    }  
    return Math.min(stop1, Math.min(stop2, stop3));  
}
```

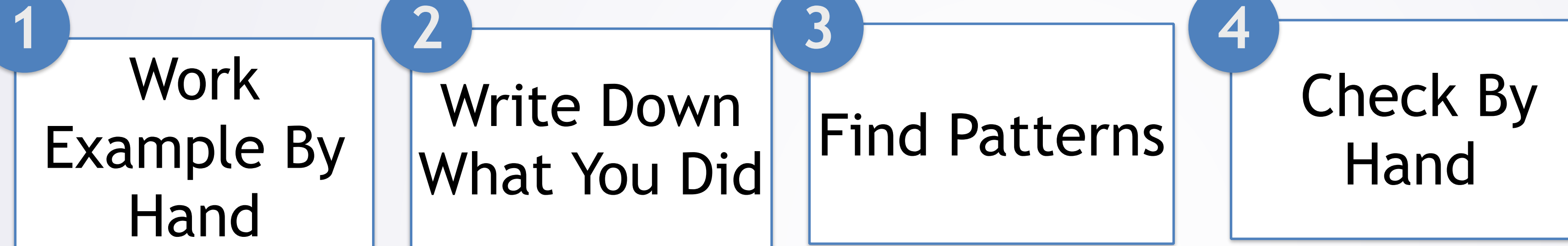
Translating Algorithm to Code

- Find each stop codon (after start)
 - If found, and multiple of 3 from start, OK
 - Choose closest of those that are OK

```
int stop2 = dna.indexOf("taa", index);  
if (stop2 == -1 || (stop2-index) % 3 != 0){  
    stop2 = dna.length();  
}  
int stop3 = dna.indexOf("tag", index);  
if (stop3 == -1 || (stop3-index) % 3 != 0){  
    stop3 = dna.length();  
}  
return Math.min(stop1, Math.min(stop2, stop3));  
}
```


The Last Three Steps

- You should check several examples before translating into code—we did this quickly



- Create loop to find all genes
 - Combine while loop with **findStopIndex**

