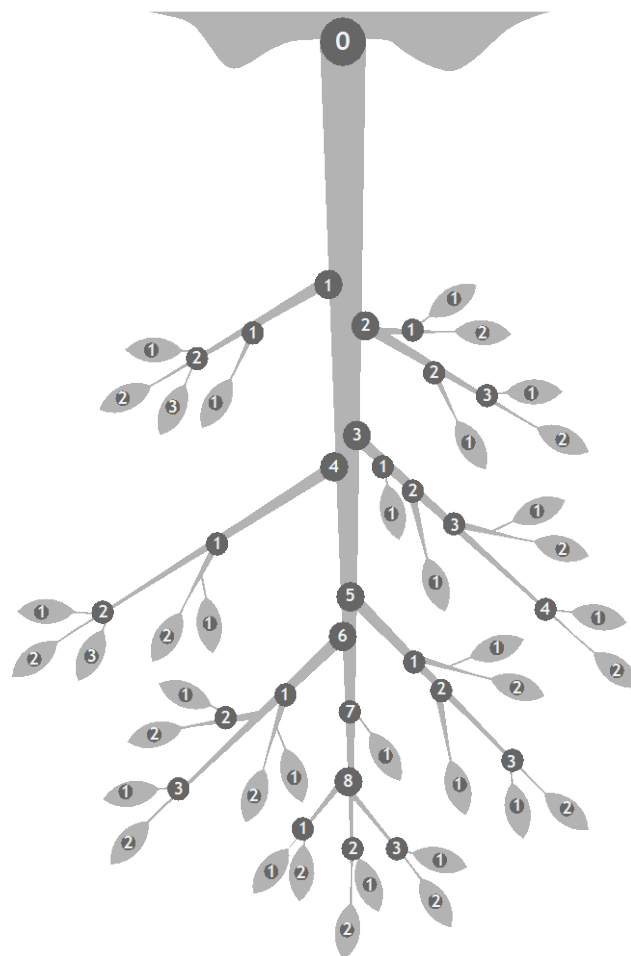
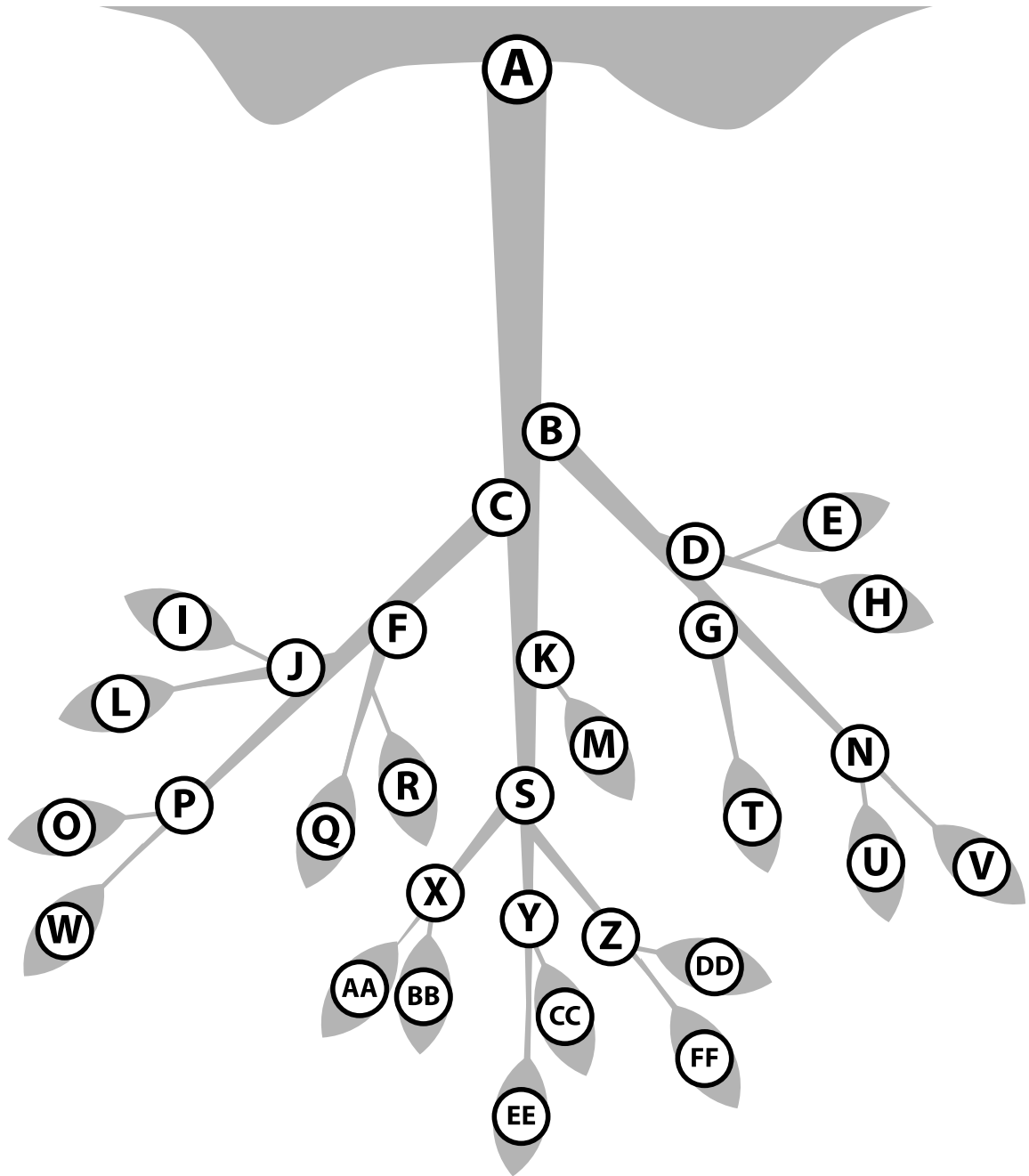


Climbing the XML Tree

Let XPath be your ladder
Exercises





NODES EXAMPLE XML

```
<A>
  <B>
    <D>
      <E>text</E>
      <H>text</H>
    </D>
    <G>
      <T>text</T>
    </G>
    <N>
      <U>text</U>
      <V>text</V>
    </N>
  </B>
  <C>
    <F>
      <Q>text</Q>
      <R>text</R>
    </F>
    <J>
      <I>text</I>
      <L>text</L>
    </J>
    <P>
      <O>text</O>
      <W>text</W>
    </P>
  </C>
  <K>
    <M>text</M>
  </K>
  <S>
    <X>
      <AA>text</AA>
      <BB>text</BB>
    </X>
    <Y>
      <EE>text</EE>
      <CC>text</CC>
    </Y>
    <Z>
      <FF>text</FF>
      <DD>text</DD>
    </Z>
  </S>
</A>
```

Answer the following questions by referring to the XML Node Bush graphic and Node Example XML.

Nodes

1. Which letter is the name of the root element node? _____
2. Which 4 letters are the names of the first level element nodes branching from the <A> node? _____
3. Which letter is the name of a second level element node branching from the <K> node? _____
4. Which 3 letters are the names of second level element nodes branching from the <C> node? _____
5. Which 2 letters are the names of third level element nodes branching from the <Z> node? _____

Relations

1. Which 4 letters are the names of the children of <A>? _____
2. Which 3 letters are the names of the children of ? _____
3. Which letter is the name of the parent of <C>? _____
4. Which letter is is the name of name of the sibling of <Q>?

5. Which letter represents the sibling of <E>? _____
6. Which 2 letters are the names of nodes without siblings?

7. Which letter represents the preceding sibling of of <Y>?

8. Which letter represents the following sibling of <Y>?

9. Which 3 letters represents the ancestors of <R>? _____
10. Which 9 letter represents the descendants of <C>?

XPATH NAVIGATION

Node Types

- Element, Attribute, Text, Namespace, Processing-instruction, Comment, Document

Node Relationships

- Parent, Child, Siblings, Ancestors, Descendants

Selectors

XPath Expression	Selects
<i>node-name</i>	All nodes with the name <i>node-name</i>
/	Root node
// <i>first-element</i>	Root element
//	Nodes anywhere in the document
// <i>node-name</i>	Nodes anywhere in the document with the name <i>node-name</i>
.	Current node
..	Parent node
@	Attribute node
*	Any element node
@*	Any element attribute
node()	Any node of any kind
text()	Text node

Axes

Name	Selects
ancestor	All ancestors (parent, grandparent, etc.) of the current node
ancestor-or-self	All ancestors (parent, grandparent, etc.) of the current node and the current node itself
attribute	All attributes of the current node
child	All children of the current node
descendant	All descendants (children, grandchildren, etc.) of the current node
descendant-or-self	All descendants (children, grandchildren, etc.) of the current node and the current node itself
following	Everything in the document after the closing tag of the current node
following-sibling	All siblings after the current node
namespace	All namespace nodes of the current node
parent	Parent of the current node
preceding	All nodes that appear before the current node in the document, except ancestors, attribute nodes and namespace nodes
preceding-sibling	All siblings before the current node
self	Current node

Predicates

XPath Expression	Selects
[1]	First element
[last()]	Last element
[position() > 2]	Elements whose positions are greater than 2
[@attribute-name]	Elements that have the attribute <i>attribute-name</i>

Operators

Operator	Description
	Computes two node-sets
+	Addition
-	Subtraction
*	Multiplication
div	Division
=	Equal
!=	Not equal
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
or	or
and	and
mod	Modulus (division remainder)

String Functions

Function	Pattern
substring()	substring(<i>string</i> , <i>start</i> , <i>len</i>) or substring(<i>string</i> , <i>start</i>)
string-length()	string-length(<i>string</i>)
contains()	contains(<i>string1</i> , <i>string2</i>)
normalize-space()	normalize-space(<i>string</i>)
starts-with()	starts-with(<i>string1</i> , <i>string2</i>)
ends-with()	ends-with(<i>string1</i> , <i>string2</i>)
substring-before()	substring-before(<i>string1</i> , <i>string2</i>)
substring-after()	substring-after(<i>string1</i> , <i>string2</i>)
concat()	concat(<i>string</i> , <i>string</i> ,...)
matches()	matches(<i>string</i> , <i>pattern</i>)
replace()	replace(<i>string</i> , <i>pattern</i> , <i>replace</i>)
tokenize()	tokenize(<i>string</i> , <i>pattern</i>)

Adapted from XPath Tutorial (https://www.w3schools.com/xml/xpath_intro.asp) by Annie Glerum October 18, 2017

NAVIGATION EXAMPLE XML

```
<records>
  <record>
    <author>
      <surname>Bock</surname>
      <given>Jerry</given>
    </author>
    <title>
      <non_sorting>The</non_sorting>
      <title_proper>Apple Tree</title_proper>
    </title>
    <publication>
      <pub_name>Columbia</pub_name>
      <pub_date>1966</pub_date>
    </publication>
  </record>
  <record>
    <author>
      <surname>Lerner</surname>
      <given>Alan Jay</given>
      <date>1918-</date>
    </author>
    <title>
      <title_proper>Camelot</title_proper>
      <subtitle>original London cast soundtrack</subtitle>
    </title>
    <publication>
      <pub_place>North Hollywood, CA</pub_place>
      <pub_name>Sarabande</pub_name>
      <pub_date>1982</pub_date>
    </publication>
  </record>
</records>
```


Answer the following questions by referring to the XPath Navigation sheet and the Navigation Example XML.

Selectors

1. Which selector would be used to navigate to the root element of the XML Navigation Example? _____
2. How many elements would **//record** match? _____
3. If the current node (.) is any **<record>** element, which selector would be used to navigate to **<records>**? _____

Axes

1. If the current node is **<author>**, which selector would be used to navigate to **<record>** and **<records>**? _____
2. If the current node is the first **<record>** element, which selector would be used to navigate to second **<record>** element?

3. If the current node is the second element **<record>**, which selector would be used to navigate to elements **<author>**, **<surname>**, **<given>**, **<date>**, **<title_proper>**, **<subtitle>**, **<publication>**, **<pub_place>**, **<pub_name>**, and **<pub_date>**?

Predicates

1. What is the value of **text()** for **records/record[1]/author/surname/?** _____
2. What is the value of **text()** for **records/record[2]/author/surname/?** _____
3. Which element does **records/record[2]/*[last()]/*[1]** navigate to? _____

Operators

1. Which predicate would be the same as **record[1 + 1]**?

2. Does **records/record[title/title_proper = 'Camelot']** navigate to the first or second **<record>** element? _____
3. Does **records/record[publication/pub_date != '1982']** navigate to the first or second **<record>** element? _____

String Functions

1. If the current node is **record[1]/title/**, what is the value of **substring(title_proper,1,3)**? [XPath indexes begin with 1.]

2. What is the path, including predicates, that matches **starts-with(subtitle,'original')**? _____
3. Which elements have values that match **matches(.,'[0-9]{{4}}')**? _____