

# Test 01 Answer Key

## Required

1) a. and c. are incorrect as they violate proper naming conventions: a) begins with a digit (names can contain digits, but they cannot begin with digits) and c) contains a space.

Some possible alternatives for a. are:

```
<Onefish> </Onefish>
<one_fish> </one_fish>
<one-fish> </one-fish>
<one.fish> </one.fish>
```

Some possible alternatives for c. are:

```
<thingTwo> </thingTwo>
<thing2> </thing2>
<thing_two> </thing_two>
<thing.two> </thing.two>
```

2) In essence, descriptive markup describes what the text inside the element *is* while presentational markup describes what the text *looks like*. The range of things a bit of text can *be* is unlimited, but the ways it can look are more limited. We prioritize descriptive markup (we mark things up according to what they are, and not according to how they look) in the digital humanities because it is easy to go in later and transform the descriptive markup into presentational markup (you will learn how to do this!), but it is much harder the other way around.

A general way to describe this difference is that it is easier to conflate things than to separate them. For example, if we tag book titles, emphasis, and foreign words according to what they are, we can render them all as italic. But if we tag them all the same way, as italic, a computer cannot easily figure why each one happened to be italicized.

3) This fragment is not well-formed because the elements `<emph>` and `<bold>` (emphasis and bold text respectively) are not nested correctly. If an element (in this case `<emph>`) is opened before another element (in this case `<bold>`), then the `</bold>` end-tag must be specified before the `</emph>` end-tag. One possible corrected fragment would be:

```
<para>My friend thinks Dr. Seuss is all nonsense, but I think something nonsensical
can <emph>still be <b>wonderful</b></emph>!</para>
```

4) The four types of content an XML element may have (see our introductory XML tutorial for more information) are:

*Element content* contains only other elements, e.g., the `<shopping_list>` element in:

```
<shopping_list>
  <item>Coffee</item>
  <item>Donuts</item>
</shopping_list>
```

*Textual content* contains only plain text, e.g., the content of the `<item>` elements above.

*Mixed content* contains a mixture of plain text and elements, e.g., the `<p>` element in:

```
<p><city>Paris</city> is the capital of <country>France</country> and
<city>Berlin</city> is the capital of <country>Germany</country>.</p>
```

*An empty element* contains no content at all, and is often used as a marker or milestone, as in:

```
<end/>
```

5. The structure of an XML document is often referred to as a tree because trees are hierarchical: the trunk splits into large branches, which split into smaller branches, etc. Each XML document must have a single root node from which all the “branches” and “leaves” emerge, e.g.:

```
<root>
  <big_branch>
    <smaller_branch>
      <leaf>(some content)</leaf>
    <smaller_branch>
  </big_branch>
</root>
```

## Extra-credit

1) For a document to be *well-formed* it must have 1) a single root element that 2) contains all its properly nested child elements (no overlapping or missing tags).

For a document to be *well-balanced* it need only fulfill the second requirement, that is, it cannot have any overlapping or missing tags, but a well-balanced document does not require a root node. Since a well-formed document has all of the requirements of a well-balanced document plus one more, a well-formed document will always be well-balanced, but a well-balanced document may not be well-formed.

2) The left angle bracket "<" and the ampersand "&" are reserved characters in XML. To represent them in text we must use *entities*, which begin with an ampersand and end with a semicolon. The entity for the left angle bracket is: `&lt;` and the entity for the ampersand is: `&amp;`. Other characters (">", apostrophe, double-quotation mark) are allowed in some XML contexts and not in others, and where they aren't allowed as literal characters, they can be represented by `&gt;`, `&apos;`, and `&quot;`.