

付録A Webページの作成

A.1 HTML

A.1.1 HTMLとは

HTMLは _____ の略で、ハイパーテキストを記述するための言語である。

- ハイパーテキスト — 互いに関連づけられた文書の集まり。関連付けのことを _____ という。

WWW上の文書（Internet Explorer（以下IE）やNetscape Navigator（以下Netscape）などのブラウザで閲覧できる文書）は、通常このHTMLで書かれている。

このプリントでは、HTMLという規格のうち、基本的なごく一部のみを紹介する。もっと詳しく知りたい人は以下のリンクや、参考書を参照すること。

- W3CのHTMLのホームページ（<http://www.w3.org/MarkUp/>）
- 初めてのホームページ講座（<http://www.hajimeteno.ne.jp/>）
- Another HTML-lint（<http://i16.jp/htmlLint/>）

A.1.2 HTML文書の作成

HTML文書はメモ帳のようなテキストエディタで作成することができる。フォント（字の形）や字の大きさ・色などを変えるときには、_____ というものを使う。HTMLのタグは「<... >」という形をしている。例えば、「太い」のように文字を太字にするには、「太い」のように書く。が太字の開始を、が太字の終了を表す。すると、ブラウザで見たときには「太い」の部分が太字で表示される。一般にHTMLのタグは<... >と</... >の対という形で使われることが多く、前者が何かの開始、後者が何かの終了を表す。（なお、タグに使うアルファベットは大文字・小文字は区別しないので、先ほどの例は太いでも構わない。ただし、新しいHTMLの規格（XHTML1.0）では _____ という規格に厳密に従うため、大文字と小文字は区別され、HTMLのタグは _____ に統一されている。）

Netscape Composerなどのソフトを使えば、ワープロのようなインターフェースで、タグを直接見ることなく、整形された形をその場で確認しながらHTML文書を作成することもできる。しかし、タグを知っていれば、他人の書いたHTML文書を見て、そのテクニックを真似することもできる。（IEの場合、メニューの「表示」「ソース」で、Netscapeの場合、メニューの「表示」「ページのソース」で、WebページのHTML文書を見ることができる。）ここでは、テキストエディタを用いてHTML文書を作成することを学ぶ。

HTML は本来、_____ではなく、_____に、設計されたものである。そのため、ワープロのように自由に体裁を整えられない場合がある。HTML 文書を作成するときは、この点を忘れないようにする必要がある。

A.1.3 全体の構造

HTML 文書は、次のような構造をしている。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ja">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-2202-JP">
    <title>ここにタイトルを入力</title>
  </head>
  <body>
    ここに本文を入力
  </body>
</html>
```

このような内容のファイルをメモ帳で作成し、たとえば test.html という名前をつけて、保存する。

HTML 文書のファイルの名前は、通常、拡張子が _____ になるようにする。

まず、最初の<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"> という部分は以下の HTML 文書が HTML 4.01 Transitional という規格に基づいていることを示す。(XHTML 1.0 の場合はこの部分は

```
<?xml version="1.0" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

となる。最初の<?xml version="1.0" ?>が XML の“お約束”である。)

残りの全体が <html lang="ja"> ~ </html> というタグで囲まれている。これは HTML 文書のお約束である。lang="ja" は文書の言語が日本語であることを示している。<head> ~ </head> の間は、_____ と呼ばれ、本文としては表示しない情報を書く。特にタイトルはこのヘッダ部分に、<title> ~ </title> に囲んで書く。このタイトルがブラウザのタイトルバーに表示される。

<meta http-equiv="Content-Type" content="text/html; charset=ISO-2202-JP"> の最後の charset=ISO-2202-JP の部分は、文字コードとして JIS コードを使用していることを示している。これは、日本語の文字化けを防ぐのに有効である。

charset	文字コード	備考
ISO-2202-JP	JIS コード	
Shift_JIS	シフト JIS コード	x-sjis は誤り。
EUC-JP	EUC JIS コード	UNIX 系のコンピュータでは標準の文字コード

Windows 系の OS で日本語の入った文書をメモ帳などで新規に作成すると、普通シフト JIS コードになるので、その場合、

```
<meta http-equiv="Content-Type" content="text/html; charset=_____ ">
```

と書くことになる。

参考リンク

- 文字コードの話 (<http://euc.jp/i18n/charcode.ja.html>) (伊藤隆幸氏)

<body> ~ </body>の間が本体である。ここにある部分が、実際に表示される。そして、以下に紹介するタグで文書の構造を明確にしたり、見かけを変えたりすることができる。タグに囲まれていない文字は普通に表示される。

A.1.4 見出し

<h*n*> ~ </h*n*> (*n* = 1 ~ 6) は _____ を指定する。 *n* は 1 ~ 6 までの数字で、1 に近いほど大きな単位の見出しになり、字が大きくなる。

入力	表示
<h1> <i>n</i> が 1の時 </h1>	<i>n</i>が 1の時
<h2> <i>n</i> が 2の時 </h2>	<i>n</i>が 2の時
<h3> <i>n</i> が 3の時 </h3>	<i>n</i>が 3の時
<h4> <i>n</i> が 4の時 </h4>	<i>n</i>が 4の時
<h5> <i>n</i> が 5の時 </h5>	<i>n</i>が 5の時
<h6> <i>n</i> が 6の時 </h6>	<i>n</i>が 6の時

A.1.5 改行・段落・区切り線

HTML 文書の中の改行は、ブラウザに表示するときは無視される。(空白と同様に扱われる。) 強制的に改行をいれたい場合は _____ というタグを使う。
には対応する終了タグはない。(なお XHTML 1.0 ではこのように終了タグのないタグは最後に / を入れて _____ のように書くことになっている。)

入力	表示
	1
1 2 3 	2 3

あいだに空行を入れて段落を作成したい場合は、<p> ~ </p>というタグで段落の文章を囲む。(p は paragraph の p である。)

また、水平線を入れて文章を仕切りたい時は _____ というタグを用いる。<hr>も
と同様、対応する終了タグはない。(XHTML 1.0 では<hr />と書く。)

A.1.6 フォント (形・大きさ・色)

字の形を変えるタグには、次のようなものがある。

タグ	意味	入力例	表示例	参考
~	太字	太い	太い	
<i>~</i>	斜字	<i>斜め</i>	<i>斜め</i>	
<u>~</u>	下線	<u>下線</u>	<u>下線</u>	将来廃止予定
<tt>~</tt>	等幅文字	<tt>WWWiii</tt>	WWWiii	ttを使わなければ WWWiii
<s>~</s>	打消	<s>打消</s>	打消	将来廃止予定

字の大きさを変えるには、~ というタグを用いる。n は 1~7 で、7 が最大である。

入力	表示	備考
n	n=1	
n	n=2	
n	n=3	標準のサイズ
n	n=4	
n	n=5	
n	n=6	
n	n=7	

字の大きさを変えるには、~ というタグを用いる。色名には次のようなものが使える。

色名	日本語	見本	備考	色名	日本語	見本	備考
black	黒		#000000	green	緑		#008000
silver	銀		#C0C0C0	lime	ライム		#00FF00
gray	灰		#808080	olive	オリーブ		#808000
white	白		#FFFFFF	yellow	黄		#FFFF00
maroon	えび茶		#800000	navy	濃紺		#000080
red	赤		#FF0000	blue	青		#0000FF
purple	紫		#800080	teal	コガモ		#008080
fuchsia	フクシア		#FF00FF	aqua	水色		#00FFFF

全てのブラウザで表示できることが保証されているのはこれだけであるが、通常は他の色名を使うことができる。さらに、赤 (R) 緑 (G) 青 (B) の光の 3 原色の強さを 00~FF までの _____ (256 段階) で指定する方法もある。~ という形で、RR, GG, BB がそれぞれ赤、緑、青成分の強さである。

- 16 進数 — 通常の数の表わし方は 10 進数で 0~9 まで 10 種類の数字を用いるが、16 進数は 0~9 に加えて A, B, C, D, E, F の計 16 種類の数字を用いる。16 進数の A は 10 進数の 10 に、16 進数の F は 10 進数の 15 に対応する。また、16 進数の 10 は 10 進数の 16 に、16 進数の FF は 10 進数の 255 (=15*16+15) に対応する。

A.1.7 CSS を使ったフォントの指定

font タグは将来廃止予定で、代わりに CSS (Cascading Style Sheet) という仕組みを用いることになっている。

CSS では、ヘッダで一括してフォントや色などのスタイルを指定することもできる (これが本来の使い方)。しかし、ここでは各タグに _____ という属性を用いてスタイルを適用する方法を紹介する。

style 属性は、<p> ~ </p> や <div> ~ </div> などのいろいろなタグに追加することができる。スタイルの適用のみを目的とするならば、 ~ というタグを使う。いくつかのスタイルをセミコロン (;) で区切って、同時に書くこともできる。

字の形を変えるスタイルには、次のようなものがある。

入力例	表示例
太い	太い
斜め	<i>斜め</i>
下線	<u>下線</u>
打消	打消
あああ	<i><u>あああ</u></i>

字の大きさを変えるには、 " _____ : " というスタイルを用いる。 _____ には次のようなものが使える。

★	表示
xx-small	xx-small
x-small	x-small
small	small
medium	medium
large	large
x-large	x-large
xx-large	xx-large

字の色を変えるには " _____ : 色名 " というスタイル、背景色を変えるには " _____ : 色名 " というスタイルを用いる。

入力例	表示例
赤	赤
青	青
青	青

A.1.8 「<」や「>」

「<」や「>」という文字は HTML 文書中で特別な意味を持っているので「<」や「>」自体を表示するには特別な形が必要である。また、HTML 中では空白はいくつ連続していても、1つの空白と同様に扱われるので、複数の空白を挿入するときにも特別な方法が要る。

入力	表示	備考
<	<	less than
>	>	greater than
 	(空白)	空白を複数個挿入したいときに用いる
&	&	ampersand

A.1.9 箇条書き・連番つき箇条書き・右揃え・中央揃え

箇条書きは _____ ~ _____ で箇条書きする部分を囲む。そして各項目を ~ で囲む。連番付の箇条書きは、 ~ の代わりに、_____ ~ _____ を使う。

入力	表示
 項目その 1 項目その 2 項目その 3 	<ul style="list-style-type: none"> • 項目その 1 • 項目その 2 • 項目その 3
 項目その 1 項目その 2 項目その 3 	<ol style="list-style-type: none"> 1. 項目その 1 2. 項目その 2 3. 項目その 3

段落の中央揃え・右揃えはそれぞれ、<div align="center"> ~ </div> と <div align="right"> ~ </div> である。

```
<div align="center">これは真ん中</div>
<div align="right">これは右</div>
```

と入力すると、

これは真ん中

これは右

のように表示される。

<div> ~ </div> の代わりに <p> ~ </p> に align 属性をつけてもほとんど同じである。ただし p タグの場合は、段落の前後が 1 行ずつ空く。

A.1.10 リンク

ブラウザで表示されている HTML 文書のある部分をクリックすると、他の文書が表示されるようにできる。この機能がリンクである。リンクを作成するタグは、

```
<a href="場所"> ~ </a>
```

である。

「場所」の部分に、この部分をクリックしたときに表示する文書のある場所を指定する。この「場所」は、HTML 文書をブラウザで表示したときに、IE の場合「アドレス」、Netscape の場合「場所」のところに表示されている文字列のことである。

例えば、

```
<a href="http://guppy.eng.kagawa-u.ac.jp/Programming2/index.html">
  プログラミング II のホームページ
</a>
```

と書くと、プログラミング II のホームページのようにリンクができる。

この場所のことを正式には _____ (Uniform Resource Locator) という。URI (Uniform Resource Indicator) ということもあるが、厳密には URI の方が URL より広い概念である。

URL には絶対 URL と相対 URL がある。

絶対 URL とは“http://guppy.eng.kagawa-u.ac.jp/Programming2/index.html”のように場所を省略することなく、すべて記述したものである。

最初の http は場所にアクセスするためのプロトコルを表す。http 以外にも ftp などのプロトコルがあるが、もっとも良く使うのは http である。

:と//の後の guppy.eng.kagawa-u.ac.jp はコンピュータ名である。(インターネットに直接つながっているコンピュータには、このようなユニークな名前がついている。)

残りの /Programming2/index.html が _____ である。つまりコンピュータの記憶装置の中の階層構造のどこにあるかを表している。Windows で用いるパスと異なり、区切りの文字は¥(バックスラッシュ)ではなく、/(スラッシュ)である。また、Windows ではパス中のアルファベットの太文字と小文字は区別しないが、URL では一般的には大文字と小文字は区別する。

相対 URL は絶対 URL の最初のいくつかの部分を省略したもので、現在の URL に対する相対的な位置を表す。

特に単にファイル名を書けば、同じ場所(ディレクトリ)にあるファイルを表す。例えば、http://guppy.eng.kagawa-u.ac.jp/Programming2/HTML.html という URL で表されているファイルから、index.html という相対 URL で参照されるファイルの絶対 URL は http://guppy.eng.kagawa-u.ac.jp/Programming2/index.html である。

下の例は現在の URL が http://guppy.eng.kagawa-u.ac.jp/Programming2/HTML.html の場合である。

相対 URL	絶対 URL
index.html	http://guppy.eng.kagawa-u.ac.jp/Programming2/index.html
Examples/template.html	http://guppy.eng.kagawa-u.ac.jp/Programming2/Examples/template.html
../ProgLang/index.html	http://guppy.eng.kagawa-u.ac.jp/ProgLang/index.html

.. は Windows の場合と同様、_____ (親ディレクトリ) を表す。

A.1.11 コメント

<!-- と-->の間は _____ になる。つまり、ブラウザはこの部分をまったく無視する。コメントは改行した時に入ってしまう空白をキャンセルするのも使うことができる。

入力	表示
AB CDE	AB CDE
AB<!-- -->CDE	ABCDE

A.1.12 整形済みテキスト

通常 HTML 文書中では空白はいくつ続けても空白 1 つと同じ扱いになり、改行は空白と同じ扱いになる。空白や改行を入力のまま表示するには、`<pre>` ~ `</pre>` というタグを使う。

入力	表示
12345 12345 12345	12345 12345 12345
<code><pre></code> 12345 12345 12345 <code></pre></code>	12345 12345 12345

A.1.13 テーブル

テーブル(表)を作るためには、まず全体を `<table border>` ~ `</table>` というタグで囲む。(枠線付きのテーブルの場合は、`border` という属性をつけて、`<table border>` ~ `</table>` のようにする。) 次に行を `<tr>` ~ `</tr>` というタグで囲む。最後に各項目を `<td>` ~ `</td>` で囲む。ただし、見出しに当たる項目の場合は、`<th>` ~ `</th>` で囲む。

入力	表示																				
<code><table border></code> <code><tr><td>1</td><td>2</td><td>3</td></tr></code> <code><tr><td>4</td><td>5</td><td>6</td></tr></code> <code><tr><td>7</td><td>8</td><td>9</td></tr></code> <code></table></code>	<table border="1"> <tbody> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>4</td><td>5</td><td>6</td></tr> <tr><td>7</td><td>8</td><td>9</td></tr> </tbody> </table>	1	2	3	4	5	6	7	8	9											
1	2	3																			
4	5	6																			
7	8	9																			
<code><table border></code> <code><tr><th>A</th><th>B</th><th>A+B</th><th>A・B</th></tr></code> <code><tr><td>F</td><td>F</td><td>F</td><td>F</td></tr></code> <code><tr><td>F</td><td>T</td><td>T</td><td>F</td></tr></code> <code><tr><td>T</td><td>F</td><td>T</td><td>F</td></tr></code> <code><tr><td>T</td><td>T</td><td>T</td><td>T</td></tr></code> <code></table></code>	<table border="1"> <thead> <tr><th>A</th><th>B</th><th>A+B</th><th>A・B</th></tr> </thead> <tbody> <tr><td>F</td><td>F</td><td>F</td><td>F</td></tr> <tr><td>F</td><td>T</td><td>T</td><td>F</td></tr> <tr><td>T</td><td>F</td><td>T</td><td>F</td></tr> <tr><td>T</td><td>T</td><td>T</td><td>T</td></tr> </tbody> </table>	A	B	A+B	A・B	F	F	F	F	F	T	T	F	T	F	T	F	T	T	T	T
A	B	A+B	A・B																		
F	F	F	F																		
F	T	T	F																		
T	F	T	F																		
T	T	T	T																		

複数行や複数列にわたるセルを作成することもできるが、エディタでいちいちテーブルのタグを入力するのは面倒なので、Netscape Composer などの HTML エディタを使用するののも一つの方法である。

A.1.14 画像

`< _____="場所" _____="代替文字列">` でその場所に画像を表示する。

「場所」は画像ファイルのある場所 (URI) で、画像ファイルの形式には JPEG 形式、PNG 形式などを使用することができる。

「代替文字列」は画像を表示できないブラウザが、画像の代わりに表示する文字列である。

A.1.15 ホームページの設置

HTML でファイルを作成したら、その HTML ファイルを Web サーバのしかるべき場所に置く。

香川大学工学部の場合、学生用の Web サーバは stwww というコンピュータである。stfile とディスクを共有しているので、stfile にファイルを置けば、stwww にファイルを置いたのと同じことになる。

stfile の各自のホームディレクトリ (Windows からは `¥¥stfile¥homes` というパスでアクセス可能) に `public_html` という名前のディレクトリを作り、そこに HTML ファイルを置く。そして、それが `foo.html` という名前のファイルならば、「`http://stwww.eng.kagawa-u.ac.jp/~s 学籍番号/foo.html`」という URL でアクセス可能になる。

なお `index.html` という名前は特別で、この場合は URL の最後のファイル名を省略することができる。上記のディレクトリに `index.html` というファイルを置いた場合は、「`http://stwww.eng.kagawa-u.ac.jp/~s 学籍番号`」という URL でアクセス可能となる。

問 A.1.1 次のテンプレートをコピーして、いろいろと書き換えて自分のホームページを作成してみよ。

ファイル名 *SanukiHanako.html*

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ja">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
  <title>Hanako Sanuki</title>
</head>
<body>
<h1 align="center">讃岐花子のホームページ</h1>
<hr>


<p> <font color="red">こんにちは。 </font> </p>

<h2>私のお勧めページ</h2>
<ul>
  <li><a href="http://www.pref.kagawa.jp/iryotandai/">香川県立医療短期大学</a></li>
  <li><a href="http://www.eng.kagawa-u.ac.jp/">香川大学工学部</a></li>
</ul>

<!-- この部分はコメントです。画面には表示されません。 -->
<hr>
<i>さぬきはなこ (hanako@eng.kagawa-u.ac.jp)</i>
</body>
</html>
```

網かけの部分は通常書き換える必要のない部分である。

A.1.16 Web ページのネチケットについて

Web ページは全世界に公開するものなので、作成する時に、いくつか注意すべき点がある。

- 機種依存文字を使わない

いわゆる半角カナ・ローマ数字・丸つきの数字など（**アイカ** **①②** **ⅢⅣ** **km** **キヨ** **元** **株**）は機種に依存する文字である。Windows では表示できるが、他の機種で表示できるとは限らない。Web ページは相手がどんな機種で読むかわからないので、機種依存文字は使ってはいけない。

- （特にトップページは）簡素にする

相手が低速な回線で見ている場合もあるので、サイズの大きな画像をトップページに使うのはよくない。Java Applet もやめた方がよい。Flash も作り方によっては、サイズが大きくなるのでよく考えて使う。フレームや表（`<table>`）もむやみに多用するのはやめたほうがよい。

- 他人のページをリンクする時はマナーに気をつける

他人のページをリンクする時は、法的には作者に連絡する必要はない。しかし、連絡を下さい、などと記述がある場合は、もちろん連絡する方がよいだろう。

ところで「リンクはご自由に」という意味で、「リンクフリー」という言葉を使っているページがあるが、これは英語としておかしい（「リンクがない」という意味になってしまう）ので使わないこと。

次のような点には注意する。

- 違法なページへのリンクは、やはり違法と判断される場合がある。また、ソフトウェアの違法コピーを許すようなリンクを作らないこと。
- リンク先のページを誹謗中傷するような書き方は、当然リンク先の作者を怒らせる。
- 他人のページがまるで自分の作ったページに見えるようなリンクの仕方もマナー違反になる。特に画像を直接リンクしたり、フレームのなかに他人のページを入れるのは良くない。

- 大学の Web サーバで商売をしない。

香川大学のコンピュータ機器は国民の税金でまかなわれているので当然である。

- アクセスの制限について

ネチケットとはすこし話題が異なるが、作成した HTML を全世界ではなく、内輪でだけ公開したい時がある。

もし、stwww に置いた HTML ファイルを香川大学外からアクセスできないようにしたい場合は、以下のような内容のファイル

```
<Limit GET>
order deny,allow
allow from 133.92.
deny from all
</Limit>
```

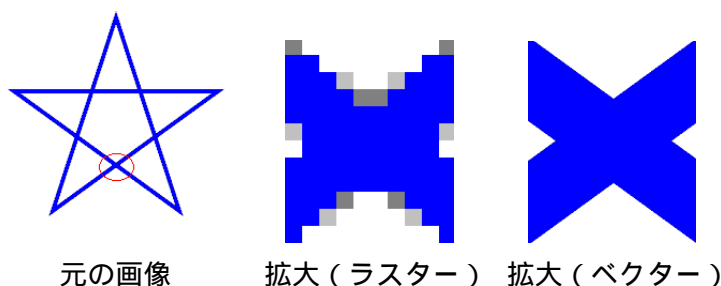
を作成し、「_____」というファイル名にして置くと、そのディレクトリ以下にある HTML ファイルは、香川大学内からのみ、アクセスできるようになる。

A.2 SVG

A.2.1 SVG とは

SVG とは、Scalable Vector Graphics の略で、2次元のベクター（ベクトル）グラフィックスのための言語仕様である。SVG も HTML と同じく W3C という団体に制定されている。

2次元画像のファイル形式は大きく _____ と _____ に分けられる。JPEG、GIF、PNG などの WWW で良く使われている画像形式は、すべてラスタ形式に分類される。ラスタ形式と言うのは、画像を縦横に分割して、格子（画素）の集まりとして考え、各画素の色を基本要素として扱う形式である。この画素の数を解像度といい、640×480, 320×240 などと表す。ラスタ形式で表されている画像の細部を拡大すると、当然この画素が目に見えるようになって、斜めの線などはギザギザになってしまう。



一方、ベクター形式というのは、画像の輪郭を直線・曲線の集まりとして捉え、線の代表点（端点・角など）の列として画像を表現する方式である。ベクター形式の場合、画像をいくら拡大しても、線の輪郭は綺麗なままである。

写真などの自然画像はラスタ形式でなければ表現することが難しいが、イラストなどの人工的な画像は、本来ベクター形式の方が保存に適している。Adobe 社の PostScript™ 形式や Macromedia 社の Flash™ が使用する SWF 形式などが代表的なベクター形式の規格である。

SVG は、W3C で仕様が策定され一般に公開されている¹。ベクター形式のファイルフォーマットで、XHTML と同じく XML の一種である。すなわちテキスト形式である、という特徴がある。

現時点では Netscape や IE などのブラウザは、標準では SVG をサポートしていないので、プラグインをインストールする必要がある。Windows 用の SVG のプラグインは Adobe の Web サイト (<http://www.adobe.co.jp/svg/viewer/install/>) または、Corel の Web サイト (<http://www.corel.com/svgviewer/>) からダウンロードできる。

まずは、SVG を使用している Web サイトを訪れて、プラグインがインストールされていることを確認する。

- Adobe のサイト (<http://www.adobe.co.jp/svg/>)
- takesato.com “SVG Developers Center” (<http://www.takesato.com/svg/>)

などを挙げておく。最後のサイトは SVG に関するリンクが豊富である。

¹<http://www.w3.org/TR/SVG/>

A.2.2 SVG ファイルの作成

試しに次のようなファイルを作成してみる。

```
<?xml version="1.0" ?>
<!DOCTYPE svg PUBLIC "-//W3C/DTD SVG 1.0//EN"
    "http://www.w3.org/TR/SVG/DTD/svg10.dtd">
<svg width="300" height="200">
  <text x="115" y="110" stroke="red" fill="orange" font-size="24">
    Hello!</text>
  <rect stroke="blue" fill="none"
    x="50" y="50" width="200" height="100" rx="35" ry="25" />
</svg>
```

ファイル名は何でも良いが、拡張子は _____ にしておく。ここでは、first.svg という名前をつけることにする。

一見するとタグ< ... >の使い方などは HTML に似ている。ただし、従来の HTML はアルファベットの
大文字と小文字は区別しなかったが、SVG は XML の一種なので大文字と小文字を厳密に _____
_____。

このファイルを Netscape または IE で表示すると次のようになる。





text は文字列を描くための命令である。なお、文字列に日本語を使う時は、注意が必要である。現時点では、Adobe SVG Viewer 3.0 は文字コードとして Unicode (UTF-8 と UTF-16) しかサポートしていないようである。Windows 2000/XP のメモ帳の場合はファイルを保存するときに「文字コード」を「Unicode」にしておく。

rect は長方形を描くための命令である。rx と ry は長方形の角のまるみを表す属性である。rect と同じような基本図形のためのタグは、この他に

<line ... />	直線を描く
<circle ... />	円を描く
<ellipse ... />	楕円を描く
<polyline ... />	折れ線を描く
<polygon ... />	多角形を描く

などがある。

上記のタグに共通の属性として次のようなものがある。

属性	意味	例	備考
stroke		stroke="red"	line は除く
fill		fill="#ffff00"	
stroke-width	線の幅	stroke-width="2"	

stroke とfill にはnone という値を与えることも可能である。その場合、それぞれ線無し、塗りつぶし無し、という意味になる。

この他にも、一次変換を行なうtransformなどの属性がある。

また各タグに特有の属性のうち良く使うものには次のようなものがある。

属性	意味	属性	意味
line の属性		ellipse の属性	
x1	始点の x 座標	cx	中心の x 座標
y1	始点の y 座標	cy	中心の y 座標
x2	終点の x 座標	rx	x 方向の半径
y2	終点の y 座標	ry	y 方向の半径
rect の属性		circle の属性	
x	左上の点の x 座標	cx	中心の x 座標
y	左上の点の y 座標	cy	中心の y 座標
width	幅	r	半径
height	高さ	polygon の属性	
rx	角の丸みの x 方向の半径	points	点列
ry	角の丸みの y 方向の半径	text の属性	
polyline の属性		x	始点の x 座標
points	点列	y	始点の y 座標
		font-size	フォントのサイズ

点列は次の例のように

$x_1, y_1 x_2, y_2 \dots x_n, y_n$

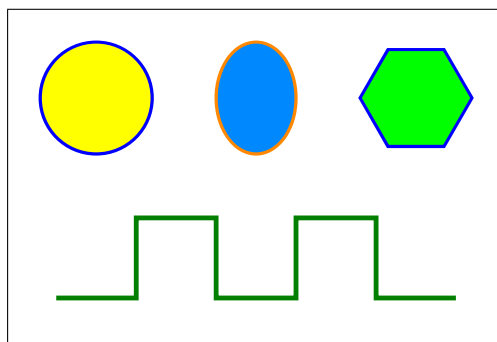
という形式で記述する。

これらのタグを使用した例をあげておく。

ファイル名 sample.svg

```
<?xml version="1.0" ?>
<!DOCTYPE svg PUBLIC "-//W3C/DTD SVG 1.0//EN"
    "http://www.w3.org/TR/SVG/DTD/svg10.dtd">
<svg width="300" height="200">
<circle stroke="blue" stroke-width="2" fill="yellow" cx="50" cy="50" r="35" />
<ellipse stroke="#ff8800" stroke-width="2" fill="#0088ff"
    cx="150" cy="50" rx="25" ry="35" />
<polyline fill="none" stroke="green" stroke-width="3"
    points=" 25,175 75,175 75,125 125,125 125,175
    175,175 175,125 225,125 225,175 275,175" />
<polygon fill="lime" stroke="blue" stroke-width="2"
    points="285, 50 267.5, 80.31 232.5, 80.31
    215, 50 232.5, 19.69 267.5, 19.69" />
</svg>
```

このファイルは次のように表示される。



また、SVG にはこのような基本図形の他にも、ベジエ (Bezier) 曲線という滑らかな曲線を描くための path 命令や、フィルタ効果、グラデーション、アニメーションなどのさまざまな機能がある。

通常 SVG は HTML で書かれた Web ページに配置されることが多いだろう。SVG 画像を Web ページに埋め込むには、HTML の _____ タグを用いる。

ファイル名 svgtest.html

```
<html>
<head><title>SVG のテスト</title></head>
<body>
SVG のテストです。<br>
<object data="logo.svg" type="image/svg+xml"
width="170" height="187">
SVG を表示できないブラウザではこの部分が出力されます。
</object>
</body>
</html>
```

object タグの data 属性に SVG のファイル名 (あるいは URI)、type 属性に image/svg+xml という文字列を指定する。width 属性と height 属性も指定する必要がある。

A.3 (おまけ) X3D

A.3.1 X3D とは

X3D とは _____ の略であり、VRML97 の後継として設計された 3D グラフィックスのファイル形式の規格である。X3D は HTML (XHTML) や SVG などと同じく、XML に基づいている。

X3D の仕様については、http://www.web3d.org/fs_specifications.htm を参照すること。また、X3D のサンプルは <http://www.web3d.org/TaskGroups/x3d/translation/examples/contents.html> で多く見つけることができる。

A.3.2 X3D 関連ソフトウェアを入手する

X3D のプラグインはいくつかあるが、現時点では BS Contact VRML (<http://www.bitmanagement.de/>) を推奨する。まずは、X3D を使用している Web ページを訪れて、プラグインがインストールされていることを確認する。次のページはインストールされているプラグインをチェックしてくれるので便利である。

- VRML Plugin and Browser Detector (<http://cic.nist.gov/vrml/vbdetect.html>)

A.3.3 スケルトン

X3D のファイルのスケルトンは次のようになる。

ファイル名 empty.x3d

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN"
    "http://www.web3d.org/specifications/x3d-3.0.dtd">
<X3D profile="Interchange">
<Scene>

    <!-- Scene graph nodes are added here. -->

</Scene>
</X3D>
```

1~4行目は XML では“お約束”の書き方である。1行目の encoding の値はここでは UTF-8 になっているが、使用する文字コードに応じて ISO-2022-JP, Shift_JIS, EUC-JP などに変更する。

5行目の X3D タグの profile の部分が、X3D の特徴的な部分で、このファイルで使用するタグの集合を指定する。Core, Interchange, Interactive, Immersive, MPEG4, Full の順に使用できるタグの種類が増える。(その代わりに再生に必要なソフトウェアが重くなる。)

<Scene> ~ </Scene> の部分 (9~11行目) に実際の 3D 物体を記述する。

X3D ファイルの拡張子は .x3d である。X3D で作成したグラフィックスを HTML ファイル中に画像として埋め込むためには、HTML の object タグを用いる。

ファイル名 x3d_test.html

```

...
<object data="first.x3d" type="model/x3d+xml"
        width="256" height="256">
  X3D を表示できないブラウザではこの部分が出力されます。
</object>
...

```

object の type 属性に用いる文字列 (MIME type) は、model/x3d+xml を指定する。

A.3.4 3D 物体

それでは、Scene に物体を追加してみよう。上記のスケルトンの `<!-- Scene graph nodes are added here. -->` の部分を次のように書き換えてみる。(以下の例ではすべてこの部分のみを示す。)

```

<Shape>
  <Cylinder />
</Shape>

```

Shape というタグの中に Cylinder というタグを入れることによって、円筒形 (Cylinder) が現れる (下図左)。



しかし、これでは色が白 (実際は明るい灰色) で立体感がほとんどない。そこでこの円筒に色をつけてみることにする。色をつけるには Appearance と Material というタグを用いる。Material タグの diffuseColor という属性に 0~1 の間の RGB 値を指定することによって、物体に色をつけることができる (上図右)。

```

<Shape>
  <Cylinder />
  <Appearance>
    <Material diffuseColor="1 0.5 0.5" />
  </Appearance>
</Shape>

```

なお、diffuseColor という属性は物体が反射する色を指定する。これに対して emissiveColor という、物体が自ら発する光の色を指定する属性もある。

Cylinder は既定値では半径 1, 高さ 2 になっている。radius と height という属性を変更することによって、この値を変更することができる。

```

<Shape>
  <Cylinder height="1.5" radius="1.8" />
  <Appearance>
    <Material diffuseColor="1 0.5 0.5" />
  </Appearance>
</Shape>

```

この結果を次図左に示す。

A.3.5 物体の移動

3D 物体はデフォルトでは中心に座標の原点が来るように配置される。物体を平行移動させたり、回転させたりするにはTransform というタグで移動させたい部分を囲む(次図右)。

```
<Transform rotation="1 0 0 0.785">
  <Shape>
    <Cylinder />
    <Appearance>
      <Material diffuseColor="1 0.5 0.5" />
    </Appearance>
  </Shape>
</Transform>
```

Transform タグには次のような属性がある。

属性名	説明	例
translation	平行移動の各座標軸方向	1 0 0
rotation	回転軸(1~3番目)と回転量(4番目)	1 0 0 0.785
center	回転と拡大・縮小の中心点	0 1 0
scale	拡大・縮小の各座標軸方向	2 1 0.5
scaleOrientation	拡大・縮小の向き(回転軸と回転量)	0 1 0 0.785

角度の単位はラジアンである。例えば translation="1 0 0"は x 軸方向へ1移動することを表し、rotation="1 0 0 0.785"は x 軸中心に0.785ラジアン(約45度)回転することを表す。なお、Transform タグで平行移動・回転・拡大が同時に2つ以上指定された時は、拡大・縮小 → 回転 → 平行移動の順に適用される。

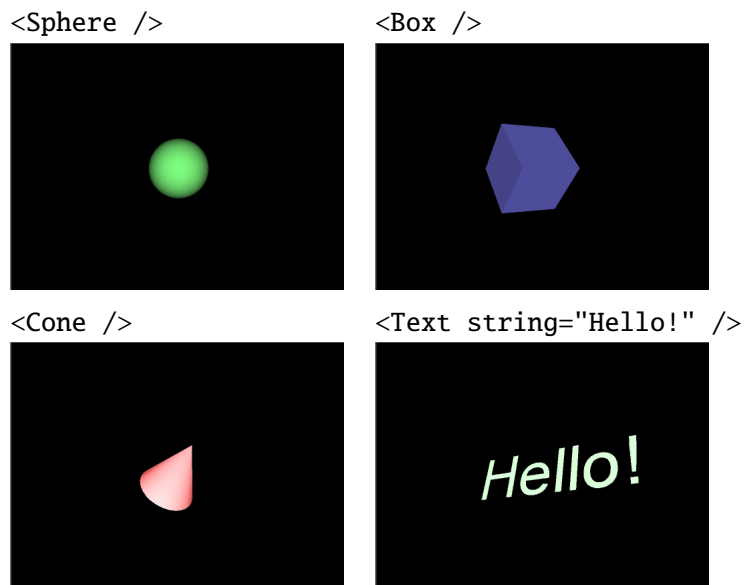


A.3.6 いろいろな3D形状

基本的な物体形状としては、Cylinder の他に Box (直方体)、Sphere (球)、Cone (円錐) などがある。また、文字列を表示するにはText というタグを用い、表示する文字列をstring という属性で与える。

```
<Text string="Hello!" />
```

次の図にこれらのさまざまな形状を適当な角度から見たものを示す。



これらのタグは次に示すような属性を指定することもできる。

タグ	属性	意味	既定値
Box	size	x, y, z 軸方向のサイズ	2 2 2
Sphere	radius	半径	2
Cone	height	高さ	2
	bottomRadius	底面の半径	1
Text	string	文字列	(必須)

さらに、ここでは詳しく説明できないが、X3Dにはこの他にも、IndexedFaceSet (複数のポリゴンを組み合わせた形状)、Extrusion (2次元の多角形に厚みを持たせた形状)、ElevationGrid (2次元座標の上で高さを指定する形状)、NurbsSurface (滑らかな自由曲面) など、多くの種類のタグがある。