**自作ホームページ「Natsu」について**

**目次**

1. はじめに
2. このホームページについて
   1. TOPページについて
   2. ABOUTページについて
   3. CONTENTSページについて
   4. PAPER&REPORTページとPROGRAMMINGページについて
   5. HISTORYページについて
   6. CONTACTページについて
3. ヘッダーについて
4. フッターについて

5.おわりに

1. **はじめに**

現在日本では少子高齢化、人口減少が進んでいる。そしてその影響がIT業界にも及んでいる。IT人材の需要が高まっている中、今後日本のI T人材は減少し、不足していくと考えられている。また、それに伴い小学生の頃からのプログラミング学習が必須となってきている。ITの仕事では、webデザイナーやエンジニア、プログラマーなどがある。プログラマーはもちろんその他のITに関する職業でも最低限のプログラミング能力は必要であると考える。つまり今後、情報社会で生きていくためにはその知識とその基礎程度は学ぶことが必要である。そこで、今後の情報社会において最低限の基礎であるホームページ作りやプログラムを勉強したいと考えた。そして今回は前者のホームページ制作をすることにした。ホームページをどのように制作したのかについて説明する。

1. **このホームページについて**

このホームページはどのようなものなのか、またどのように製作したのかについて、べる。なお第2節ではヘッダーとフッターについては取り扱わないものとする。また、私はひとつのHTMLに対してひとつのCSSを作成した。なぜなら複数のHTMLに一つのCSSにすると、そのどれかのHTMLにCSSによる影響で不具合が出た場合に対応することが容易になるからである。

* 1. **TOPページについて**

砂浜のスクリーンショット

自動的に生成された説明

図1 TOPページ 出典：筆者作成

今回製作したH P「Natsu」は白と黒を使って見やすくしたとてもシンプルなデザインである。初めて製作したホームページであり、ゼミでのレポートやプログラムを載せるホームページであるので、複雑なものは避けなるべくどこに何があるのかがわかるものにした。まずHTMLソースコードについて解説していく。このTOPページのメインのHTMLソースコードは以下の通りである。

<!--メインここから-->

<div class="main">

<p><img src="img/IMG\_1.jpeg" alt="海"＞</p>

<section>

<h2>MASSAGE</h2>

<p>このサイトでは、大学での論文やレポートをアップロードしていきます。<br>

よろしくどうぞ。</p>

</section>

</div>

<!--メインここから-->

まず、このTOPページでは見出しと文章の部分を記述した。ホームページで見出しを表示するためには、h要素を使う。h要素は見出しの意味があるタグでありこれにはh１からh6まで6段階のレベルがある。h1要素が最も大きな見出しでありそれについでh2,h3,h4…h6という段階になりh6が最も小さな見出しを表している。今回のHTMLソースコードでは<h2>~</h2>要素を使っている。次に使われているのはp要素である。p要素で囲まれた部分は段落を表し、文章だけでなく、写真やイラストなども段落として見なすことがある。このpタグで囲まれている間に<br>というものがある。これは、br要素といいbrタグを記述するとウェブブラウザで開いたときに強制的に改行することができます。このbrタグは先ほどのh要素やp要素などと違い囲まずに単独で使用しているのが特徴である。このような要素を空要素という。他には、<meta>がこの空要素に含まれる。そして、見出しと文章を記述した後この海の写真をホームページに埋め込んだ。写真をホームページに埋め込むためにはimg要素に属性を付与することが必要である。例えば属性なしでimg要素だけで表した場合はここに何か画像が埋め込まれているということしかわからずどのような画像が表示しているかまではわからない。しかし、img要素に属性を付与することによって、実際に画像を表示することができる。具体的にいうと、imgタグにsrc属性とalt属性を追加する。src属性を追加することで画像ファイルの設置場所(パス)を指定する。alt属性は基本的には省略することはできない。属性には、属性名と属性値を対の形で記述する。今回の場合、srcとaltが属性名であり"img/IMG\_1.jpeg"と"海"が属性値である。そしてこの<img src="img/IMG\_1.jpeg" alt="海"＞の意味は「imgフォルダーの中にあるIMG\_1.jpegという画像を埋め込み、もし画像が表示されなければ『海』というテキストを表示する」という意味がある。そして今回はこの<img　src="img/IMG\_1.jpeg" alt="海"＞をp要素で囲み段落として設定している。そして全体としてmainタグを使って囲み、メインコンテンツを表している。またsectionタグは「情報のカタマリ」を表すことに使い、見出しとその内容をまとめるときに使う。しかし今回、mainタグはdivタグで囲まれている。divタグは他のタグとは違いそれ自体で意味を与えることはない。div=複数の要素をグループ化するためのタグである。さらに今回はdiv要素にclass属性の値として記述した。これは、cssで記述する際にclassセレクタの方が個別に指定することが容易になり楽になるからである。また、このメインコンテンツは<!-- --!>に囲まれている。これは、HTMLにメモをするときに使うものである。この<!-- --!>で囲まれた文章はHTMLに影響を与えることはなく、メモとして使うことができる。また、これを使うことでどこに何があるのかが見やすくなるので何かしら不具合があったときに、そこを見つけるのが早くなる。次に、このHTMLに対応するCSSソースコードは以下の通りである。

.main{

width:100%;

margin-right:auto;

margin-left:auto;

text-align:center;

}

img {

border-radius:30px;

width: 100%;

}

section{

box-shadow:0 0 20px 0 #ccc;

}

まず.mainを記述した。これはclassセレクタであり、class名の前に「.」をつけたものである。.mainには、widthプロパティ、margin-rightプロパティ、margin-leftプロパティ、text-alignプロパティをを入力した。まず、widthプロパティであるがこれは、幅を指定するのである。今回、値が100%と指定しているがcssではpx, em, rem,%などがある。私は、初めてホームページ制作をしたのであまりpxやemといった単位に慣れてなく100%にすればそのホームページの最大幅に設定することができるので%を指定して製作した。次にmargin-rightプロパティとmargin-leftプロパティである。この値は、autoを指定した。autoを指定することで、ブラウザが自動的に最適な余白を付ける。これらは、pxを指定することで具体的な値を設定することができる。そしてimgにはborder-radiusプロパティとwidthプロパティを入力した。border-radiusプロパティで指定できるものとは、四隅に配置する丸み(円)の半径である。このborder-radiusプロパティを使ってimg要素の角を6px丸めた。最後sectionにはbox-shadowプロパティを入力した。box-shadowプロパティとは、要素に影のような効果をつけるものである。box-shadowプロパティの値には、「右方向の距離」、「下方向の距離」、「ぼかし距離」、「広がり距離」、「影の色」の５つの値を組み合わせて指定することができる。今回は、影を右と下にも伸ばさず(0)、ぼかし距離(10px)と影の色(グレー)を指定した。これをsection要素の周囲に影が広がるような効果になる。

**2.2 ABOUTページについて**

モニターに映ったウェブページのスクリーンショット

自動的に生成された説明

図2 ABOUTページ 出典：筆者作成

図2は作成したホームページのABOUTページであり、このメインのHTMLのソースコードは以下の通りである。

<!--メインここから-->

<div class="main">

<h1><span class="under">ABOUT</span></h1>

<p><img src="img/IMG\_2.jpeg" alt="海"></p>

<table>

<tr>

<th>Name</th>

<td>Natsu</td>

</tr>

<tr>

<th>Birthday</th>

<td>7.12</td>

</tr>

<tr>

<th>Hometown</th>

<td>Saitama</td>

</tr>

<tr>

<th>Hobby</th>

<td>Listening to music</td>

</tr>

</table>

</div>

<!--メインここまで-->

まずABOUTページでは見出しをh要素で記述した。ここでspanタグを使っているがこれは、単体では特に意味を持たないタグであるが、<span>～</span>で囲むことにより一つのグループとして扱うことができるタグである。これを使う理由はこののちに記述する。次にp要素で段落として写真を埋め込んだ。ABOUTページは、自己紹介文にあたるので左に写真、右にプロフィールを書こうと考えた。そこで使ったのは、table、tr、th、td要素である。tableタグで囲まれた部分は、それが1つの表であることを表す。tdタグで囲まれた部分は表のデータセル(マス)であり、見出しのセルはtdではなくthタグで囲む。そしていくつかのth、td要素をtrタグで囲むことで、それらのth、td要素が一業の中に並んでいることを示す。今回の場合は2列×3行の表を製作した。まず、tableタグを入力した。次にtrタグを入力し行を追加した。そしてthタグを入力し自己紹介の見出しであるName、Birthday、Hometown、Hobbyを記述した。そしてtdタグを入力し、見出しに対応する中身を記述した。次に、このHTMLに対応するCSSソースコードは以下の通りである。

img{ height: 540px;

width: 540px;

margin-left: 60px;

float: left;

}

table{margin-top:150px;

margin-left:650px;

font-size:180%;

}

.main{

width:100%;

margin-right: auto;

margin-left: auto;

}

.under{ background: linear-gradient(transparent 70%, #c0c0c0 30%);

}

まずimgを記述した。imgにはheight、width、margin-left、floatプロパティを使用した。heightプロパティを使うと、要素の高さを指定することができる。この値を540pxとして指定したので、img要素の高さが540pxになるようにした。widthプロパティでも値を540pxを指定することで写真を540px×540pxの形にした。次に、margin-leftを使って左に60pxの空白を開けた。最後にfloatプロパティを使用した。floatプロパティは、要素を左または右に寄せて、後に続く要素を開いたスペースに回り込ませるためのプロパティである。今回は写真を左に寄せるために使用した。左に寄せるために使った値はleftを使った。次はtableである。tableではmargin-top、margin-leftプロパティを使ってそれぞれ上と左に150px、650pxの空白を製作した。そして表の文字のサイズを変更するためにfont-sizeプロパティを使い180%の値を入力し、文字を1.8倍に変更した。クラスセレクタの一つのmain要素では、widthプロパティで幅を100%に設定し、margin-right、margin-leftプロパティを使い左右の空白を自動的にブラウザが決めてくれるautoを入力した。もう一つのクラスセレクタであるクラス名underである。これにはbackgroundプロパティを使って、蛍光ペンのような線を引こうとしたのでlinear-gradient()関数を使用した。lこれの仕組みは、linear-gradient(transparent　色の割合, 蛍光ペンの色 色の割合)となっている。transparentは透明を表し、今回の場合は、transparentを使い上から70%を透明、蛍光ペンの色をグレー(#c0c0c0)に設定し下の30%を#c0c0c0にした。ここで重要なのが先ほど触れたspanタグである。このタグを使ってインライン要素を囲んでいるので該当部分のみこのback-groundタグの効果を得ることができる。もし使わなければ、指定せずABOUTの文字の部分だけでなくその線が延長されて画面に横の線が入ってしまう。そのためにHTMLでspanタグを使い配置したい部分だけにback-groundプロパティを反映させた。

**2.3CONTENTSページについてスクリーンショットの画面

自動的に生成された説明**

図3 CONTENTSページ 出典：筆者作成

図3は作成したホームページのCONTENTSページであり、このメインのHTMLのソースコードは以下の通りである。

<h1><span class="under">CONTENTS</span></h1><br><br>

<ul>

<h2><span class="under">PAPER&REPORT</span></h2>

<div class="list">

<li><a href="PAPER&REPORT.html">春レポート</a></li><br><br><br>

</div>

<h2><span class="under">PROGRAMMING</span></h2>

<div class="list">

<li><a href="PROGRAMMING.html">更新中</a></li>

</ul>

</div>

CONTENTSページでは、論文やレポート、プログラミングを掲載するためのページである。まずCONTENTSページでは大見出しであるCONTENTSという文字をh1要素で記述した。 次に中見出しであるPAPER&REPORTとPROGRAMMINGをh2要素で記述した。見出しには全てback-groundプロパティを配置する予定なので全てspanタグで囲った。そして次にul要素とli要素を記述した。これらの要素は箇条書きにしたい項目の前後をまとめてulタグで囲むと、その範囲が箇条書きとして設定される。具体的にはul要素がリストを作成するためのタグであり、li要素はリストの項目を作成するタグである。箇条書きにしたい項目は、一つずつliタグで囲んだ。これを使った理由は保オームページのサイト内を回遊するためのナビゲーションメニューも箇条書きで表現するからである。今回の場合は、春レポートをクリックしたらPAPER&REPORTのHTMLのページに移動、更新中をクリックするとPROGRAMMINGのHTMLのページに移動するようにした。まずリストを作成するためのul要素タグを入力し、リスト項目を作成するli要素タグを記入した。次に、このHTMLに対応するCSSコードは以下の通りである。

.under{ background: linear-gradient(transparent 70%, #c0c0c0 30%);

}

.list{ margin-left: 40px;

}

クラスセレクタのunderというクラス名は、2.3節も解説し今後も出てくるが省略していく。今回、HTMLを記述していく上でdiv要素のクラス名を同じlistとした。その理由は、クラスセレクタのクラス名listとして設定することにより、同じようにCSS反映させたいからである。このように同ページで同じデザインやレイアウトをしたい場合クラスセレクタは非常に便利である。

**2.4 PAPER&REPORTページとPROGRAMMINGページについてスクリーンショットの画面

自動的に生成された説明** 図4 PAPER&REPORTページ 出典：筆者作成**モニターに映ったウェブページのスクリーンショット

自動的に生成された説明** 図5 PROGRAMMINGページ 出典：筆者作成

PAPER&REPORTページとPROGRAMMINGページは名前の通り、ゼミでのレポートやプログラミングしたものを掲載するためのページである。PAPER&REPORTページとPROGRAMMINGページは、図4と図5を見てもらうとわかる通り見出しとリストの文字以外同じ形なので二つまとめて解説していく。この二つのHTMLソースコードは以下の通りである。

[PAPER&REPORTページのHTMLソースコード]

<h1><span class="under">PAPER&REPORT</span></h1><br><br>

<ul>

<li><a href="自作ホームページ「Natsu」について.docx">自作ホームページ「Natsu」について</a></li>

</ul>

[PROGRAMMINGページのHTMLソースコードについて]

<h1><span class="under">PROGRAMMINFG</span></h1><br><br>

<ul>

<li><a href="作成中.docx">作成中</a></li>

</ul>

この二つのHTMLソースコードを比較してみると表示される文字だけが同じでそれ以外の要素は全て同じであることがわかる。当たり前のことだが、同じHTMLソースコードを使えば同じようにサイトでは表示される。この二つで工夫というほどではないが、2.3節で解説した文字に蛍光ペンで線を引くように表示されるback-groundプロパティを使うため、HTMLやCSSのソースコードが容易になるように同じクラス名underを使用した。次に、この二つのCSSソースコードであるがこれは2.3節で説明したback-groundプロパティを二つとも全く同じ形で使用しているため、省略する。

**2.5 HISTORYページについてスクリーンショット, 抽象, モニター が含まれている画像

自動的に生成された説明** 図6 HISTORYページ 出典：筆者作成

図6は作成したHISTORYページである。以下がそのHISTORYページのメインのHTMLソースコードである。

<h1><span class="under">HISTORY</span></h1><br><br>

<table width=80% border="1" align="center">

<tr>

<th bgcolor="#5a5a5a">DATE</th>

<th bgcolor="#5a5a5a">UPDATE</th>

</tr>

<tr>

<td>2020.5.8</td>

<td>TOPページをアップロード</td>

</tr>

<tr>

<td>2020.6.1</td>

<td>ABOUTページをアップロード</td>

</tr>

<td>2020.6.10</td>

<td>CONTENTSページをアップロード</td>

</tr>

<tr>

<td>2020.7.2</td>

<td>PAPRER&REPORTぺージをアップロード</td>

</tr>

<tr>

<td>2020.7.2</td>

<td>PROGRAMMINGページをアップロード</td>

</tr>

<tr>

<td>2020.7.11</td>

<td>HISTORYページをアップロード</td>

</tr>

<tr>

<td>2020.7.24</td>

<td>CONTACTページをアップロード</td>

</tr>

</table>

HISTORYページでは、更新履歴を表示させるためのものである。HISTORYページは、他のページと比較して違う部分はHTMLソースコードでwidthプロパティやborder属性などを使ったことである。<table width=80% border="1" align="center">の部分の意味は、tabale要素の部分をwidthプロパティで幅を80%に指定した。border="1"とは表の枠線を表示するものである。もし枠線非表示にしたければborder属性を0にすれば良い。また、枠線の太さを変えたければ1以上の値を選べば数値が大きいほど太くなる。align属性は配置を指定するもので今回は真ん中に設定したかったのでcenterとした。次に、br要素だが、これはTOPページでも説明したが<br>要素を使用した。br要素は重ねたらその分だけ改行できることがわかったのでh1要素の後に二連続で使用した。表を作るには、tabele、th、td、tr要素を使う。これは2.3節のABOUTページで説明したので省略する。ここで新しく使ったのは、テーブル全体の背景色を指定するbgcolor属性をつけた。今回は見出しのth要素の部分だけ色を設定したかったので<th bgcolor="#5a5a5a"></th>と記述し、濃いグレーを背景にした。CSSは同じなので省略する。

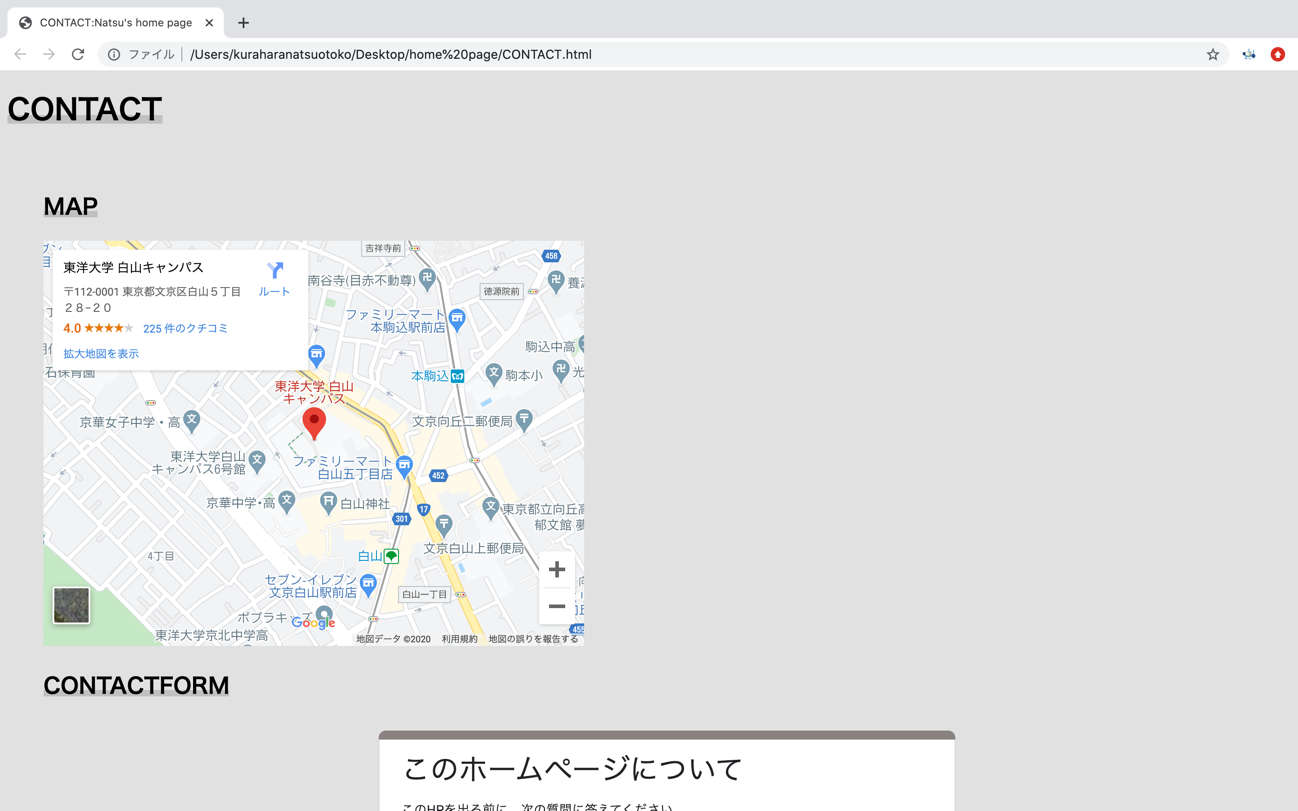
**2.7　CONTACTページについて**図7 CONTACTページ 出典：筆者作成

図7は作成したホームページのCONTACTページであり、このメインのHTMLのソースコードは以下の通りである。

<div class="main">

<h1><span class="under">CONTACT</span></h1><br>

<ul>

<h2><span class="under">MAP</span></h2>

<iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3239.1137978058587!2d139.74813961528858!3d35.72341973526435!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x60188db7b2c0766d%3A0xe80e1b6e1bac88a5!2z5p2x5rSL5aSn5a2mIOeZveWxseOCreODo-ODs-ODkeOCuQ!5e0!3m2!1sja!2sjp!4v1595518659767!5m2!1sja!2sjp" width="600" height="450" frameborder="0" style="border:0;" allowfullscreen="" aria-hidden="false" tabindex="0"></iframe>

<h2><span class="under">CONTACTFORM</span></h2>

<!--お問い合わせフォーム-->

<iframe src="https://docs.google.com/forms/d/e/1FAIpQLSfYlX4D\_kK7c4EF4h3-oW9tnYVC\_ho9-9c4vQL3tc9Ny5k4BQ/viewform?embedded=true" width="100%" height="800" frameborder="0" marginheight="0" marginwidth="0">読み込んでいます…</iframe>

</div>

まず、CONTACTページはMAPとCONTACTFORMを表示するためのページである。このページではMAPを表示するためにgoogleマップを埋め込み、CONTACTFORMを表示するためにgoogleフォームを埋め込んだ。googleマップをホームページで表示するためにはgoogleマップを開き、目的地である東洋大学白山キャンパスの住所を入力した。googleマップでメニューを開き「地図を共有または埋め込む」を選択し、埋め込みコードを選択しコピーした。それをHTMLに貼り付けるだけでホームページ表示されるようになる。次に、googleフォームであるが、これをお問い合わせフォームとしたのも手軽だからである。埋め込むだけで表示することができ、一からお問い合わせフォームを作るよりも早くできる。googleフォームを利用するにはgoogleアカウントでログインし、埋め込みたいフォームを選択し、フォーム上部の歯車マークをクリックして、全般タブの「回答を一回に制限する」のチェックを外さなければ回答する人がgoogleアカウントにログインしてなければ回答することができなくなる。そして上部の「送信」ボタンをクリックし、「<>」のアイコンをクリックすることでHTMLコードが表示されるのでこれを貼り付けるだけで表示された。そして、widthとheightの値を変えて大きさを合わせた。CSSソースコードは、同じなので省略する。

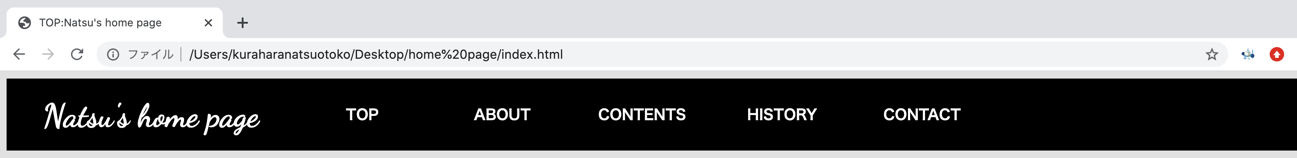
**3.ヘッダーとフッターについて**図8 ヘッダー 出典:筆者作成

図８は作成したホームページのヘッダーであり、このHTMLのソースコードは以下の通りである。

<div class="header-logo">Natsu's home page</div>

<nav>

<ul class="menu">

<li><a href="index.html">TOP</a></li>

<li><a href="ABOUT.html">ABOUT</a></li>

<li><a href="CONTENTS.html">CONTENTS</a>

<ul>

<li><a href="PAPER&REPORT.html">PAPER&REPORT</a></li>

<li><a href="PROGRAMMING.html">PROGRAMMING</a></li>

</ul>

</li>

<li><a href="HISTORY.html">HISTORY</a></li>

<li><a href="CONTACT.html">CONTACT</a></li>

</ul>

</nav>

私はヘッダーでドロップダウンメニューを作った。ドロップダウンメニューはJavaScriptが必要であると考えるかもしれないがHTMLとCSSだけで作ることが可能であった。上記のHTMLソースは、ul要素とli要素で作る標準的な2階層のリスト構造である。ul要素で1つの階層を作り、li要素で1つの項目を作ります。li要素の内側にul要素を含めることでその項目に属する2階層目を作れる。なお、li要素内の項目名はa要素を使ってリンクにすることでクイックすることでそのページに移動できるようにした。

CSSで装飾するために、最初のul要素だけにclass名menuを付加した。ドロップダウンメニューでは、HTMLソースコードはシンプルなものに見えるがCSSコードは複雑になった。以下がそのCSSソースコードである。数が多いため部分ごと解説していく。

ul.menu {

width: 100%;

margin: 0px;

padding: 0px 0px 0px 15px;

background-color: black;

}

これはメニューバーの装飾である。widthプロパティの解説はホームページでの解説の時にしたので省く。marginプロパティで、メニューバーの外側の余白を0にした。次にpaddingプロパティを使用した。値を入力する時、上下左右、個別で指定したい場合は:上 右 下 左;の順番で指定することができる。今回は左に15pxの余白をメニューバーの内側に設定した。最後にbackground-colorでバーの背景色を黒に指定した。

ul.menu li {

width: 150px;

display: inline-block;

list-style-type: none;

position: relative;

}

ul.menu a {

background-color: black;

color: white;

line-height: 80px;

text-align: center;

text-decoration: none;

font-weight: bold;

display: block;

}

ul.menu a:hover {

background-color: #404040;

color: white;

}

これは、メインメニュー項目の装飾の三つのCSSソースコードである。ul.menu liでは、widthプロパティでメニュー項目の横幅を125pxに設定し、displayプロパティには値inline-blockを指定することによりリスト項目を横並びにした。list-style-noneプロパティにnoneを指定してリスト項目の先頭の黒丸を消した。positionプロパティは、必ずrerativeに指定しなければならない。なぜなら、サブメニューの表示位置を絶対位置(absolute)で記述するためである。次のul.menu aではbackground-colorプロパティでメニュー項目の背景色を黒にし、colorプロパティでメニュー項目の文字の色を白にした。また、line-heightプロパティでメニュー項目のリンクの高さを80pxに設定した。ここの高さが小さすぎるとクリックしづらくなるので少し太めにした。text-alignプロパティでメインメニューの文字列の配置を中央に寄せ、text-decorationプロパティではメニュー項目の装飾を設定できるので下線を消す。font-weightでboldの値を設定し太字にした。displayプロパティでblockを指定することにより、文字のない余白部分もクリック可能にし、メニュー項目内全体がクリックでいるようになった。ul.menu aでhoverを指定することでメニュー項目にマウスが乗った時の色を指定することができるようになるのでbackground-colorプロパティでメニュー項目にマウスが載ったときの背景色をグレーに設定し、colorプロパティメニュー項目にマウスが載ったときの文字の色を白にした。

ul.menu li:hover ul {

display: block;

}

これはサブメニューがある場合に開く処理であり、この記述はサブメニューが1階層しか存在しない場合のものである。ul.menu li:hover ulでは、displayプロパティをblock指定することでマウスポインタが乗っている項目の内部にあるリストを表示するようにした。

/\* ▼サブメニューの装飾 \*/

/\* -------------------- \*/

ul.menu ul {

margin: 0px;

padding: 0px;

display: none;

position: absolute;

}

これはサブメニューの装飾である。ul.menu ulでは、marginプロパティでサブメニュー外側の余白をゼロにし、paddingプロパティではサブメニュー内側の余白もゼロにした。displayプロパティでは、値をnoneに設定した。noneにしなければ、標準時でも表示されるようになってしまうので、noneにして非表示にした。positionプロパティで値はabusoluteを指定し絶対配置にしなければならない。なぜなら、サブメニューは、それが属するメインメニュー項目の真下に表示されないと、サブメニューらしくは見えないので、表示場所を任意の場所にするためである。絶対配置の基準になる要素は、この要素よりも上位(浅い)階層にある「positionプロパティに値relativeが指定されている要素」あるので先に解説したここでは、「メインメニュー項目」が該当するようになる。

ul.menu ul li {

width: 150px;

border-top: 1px solid white;

}

ul.menu ul li a {

line-height: 80px; /\* サブメニュー1項目の高さ(35px) \*/

text-align: left; /\* 文字列の配置(左寄せ) \*/

padding-left: 5px; /\* 文字列前方の余白(5px) \*/

font-weight: bold; /\* 太字 \*/

}

ul.menu ul li a:hover {

background-color: #404040;

color: white;

}

これらはサブメニュー項目の装飾である。ul.menu ul liではwidthプロパティでサブメニュー１項目の横幅を150pxの値を指定した。また、border-topプロパティで: 1px solid white;と指定することで項目上側の枠線を白色で1pxの実線になる様にした。ul.menu ul li aではline-heightをメインメニューと同じ高さである80pxにして、text-alignプロパティではleftを指定することにより文字列の配列を左寄せに配置した。また、padding-leftプロパティでは文字列前方の余白を5pxに指定し、font-weightプロパティではboldでふと文字にした。ul.menu ul li a:hoverではメインメニューと同じ様にサブメニューにマウスが載ったときの背景色をグレー、文字色を白色に指定した。以上が今回製作した、ドロップダウンメニューを作るCSSの記述である。

.header-logo {

float: left;

font-size: 36px;

padding: 20px 40px;

font-family: 'Dancing Script', cursive;

color: white;

これは、ヘッダーロゴのスタイルを変えるためのCSSソースコードである。クラス名.header-logoではfloatプロパティで要素を左寄せにして、font-sizeプロパティを36pxにした。paddingプロパティでは、値を2つ設定した。この場合「上下」「左右」の順に適用される様になる。font-familyプロパティは、文字のテキストをデザインするためのものである。最後にcolorプロパティでは、白色を指定した。

4. フッターについて

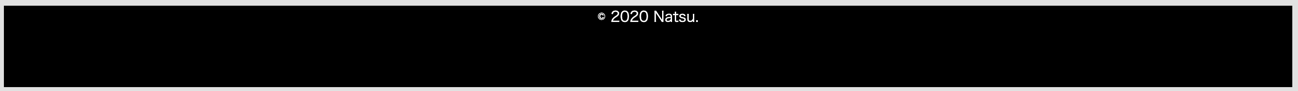
図9 フッター　出典:筆者作成

図9は作成したホームページのフッターでありこのHTMLのソースコードは以下の通

である。

<div class="footer">

<p>&copy; 2020 Natsu.</p>

</div>

フッターではクラス名footerとして製作した。p要素を使い、&copy;2020 Natsu.と記述した。&copyとすることでホームページ上ではコピーマークが表示される様になる。以下はフッターのCSSソースコードである。

.footer {

width: 100%;

　　　background-color: black;

height: 90px;

color:white;

text-align:center;

margin-right:auto;

margin-left:auto;

bottom: 0;

position: absolute;

}

ここで新しく設定したのはbottom プロパティである。これは、下からの配置位置を指定することができ、今回はフッターなのでゼロにした。また、positionプロパティでabsoluteを指定することにより、常に下に配置される様にした。

1. **おわりに**

ホームページ制作をして、HTMLとCSSを記述したがネットや本を参照することによって基本的なwebページの作成方法、リンクを使った連携や表の作り方、ファイルやMAP、CONTACTFORMの埋め込みができる様になった。今回製作して考えたことは、結局事前のデザインを考えることが重要であると考えた。当たり前ではあるが、自分に製作する技術があれば、自分が考えたデザインの通りに最終的にはたどり着く。最初のデザインがよければ、結果として良いものになる。反対にデザインが悪ければ、完成したものはデザインが悪くなる。つまり、結局自分のアイデアが重要になる。もちろん今回最低限のHTMLとCSSを記述する技術を得れたのかもしれないが、最初にデザインするもののアイデアがなければそもそも使うことがない。今回ホームページ制作をして、0から1を生み出す力が今後重要になってくると考えた。しかし、それはHTMLやCSSという最低限の記述できる技術は要らないということではない。最低限の技術がなければできることもわからないので今回学んだ技術は無駄ではない。今回学んだHTMLとCSS以外にもプログラミング言語は存在するので他の言語も最低限の技術は学びたいと考えた。

参考文献

千貫りこ(2017). 『デザインの学校　これからはじめるHTML&CSSの本[Windows10&macOS対応板]』株式会社技術評論社.

WEB色見本 原色大辞典(1997). < https://www.colordic.org/> (2020年5月8日参照).

西村文宏. 『ドロップダウンメニューをHTML＋CSSだけで作る方法』 < https://allabout.co.jp/gm/gc/23911/ >(2020年5月8日参照).

Chrono Drive. 『spanタグとは』<https://htmlcoding.co.jp/annex/dictionary/html/span/>(2020年6月1日参照).

表を作る. < http://www.yasudaya.co.jp/pworld/help/table.html> (2020年7月11日参照).

創kenブログ(2019). 『GoogleフォームをWEBサイトに埋め込む方法』 < https://souken-blog.com/2019/05/28/gogle-form/>(2020年7月24日参照).