



# PROGETTO HTML & CSS

PRESA  
TANZIONE DI EMANUELE ROMANO

Powered  
For



start2impact  
UNIVERSITY

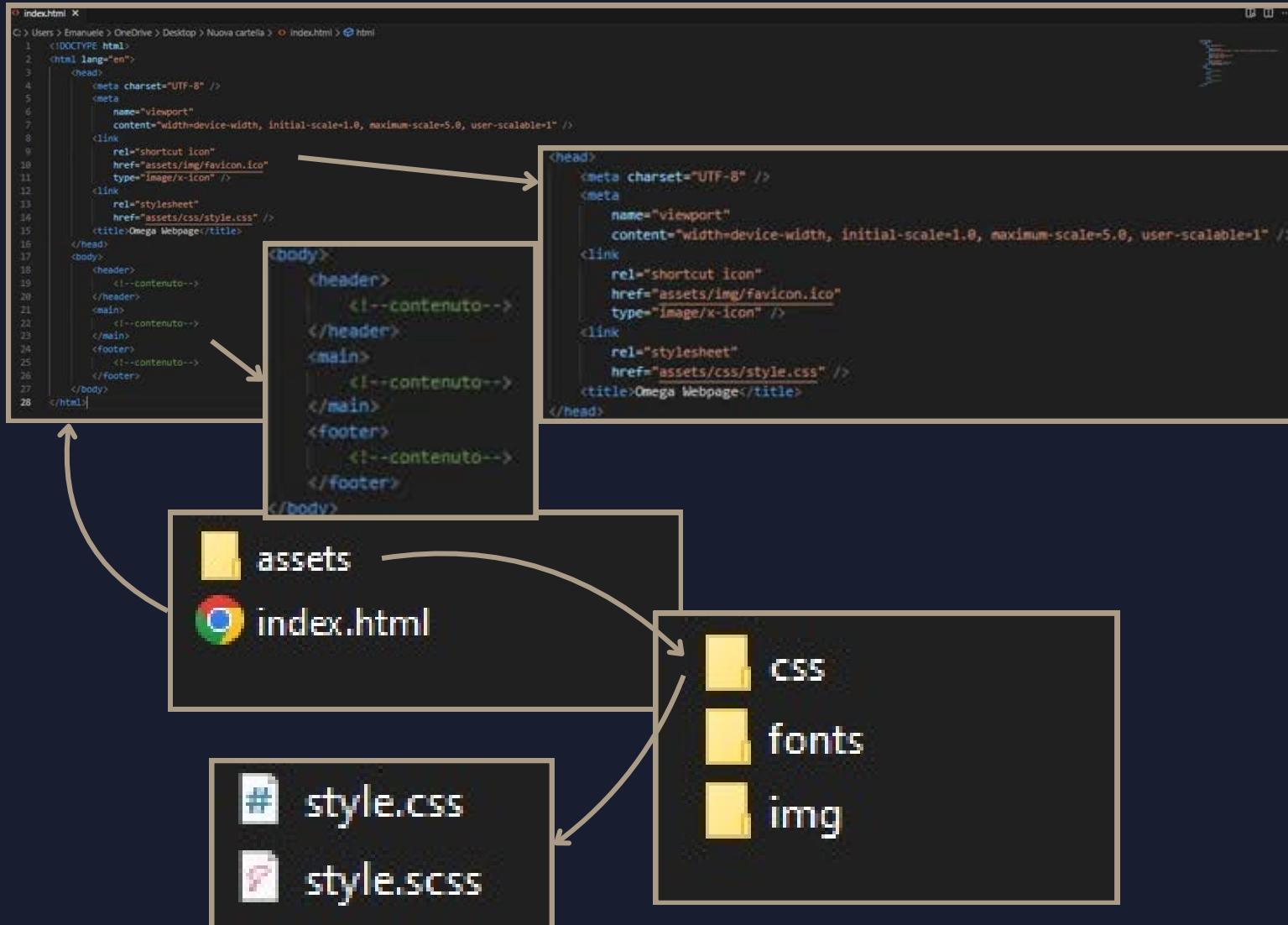
# PRIMO APPROCCIO

Per prima cosa, ho impostato il file come codice HTML digitando ! e ho aggiunto gli eventuali tag base in `<head>` e dividendo `<body>` in `<header>`, `<main>` e `<footer>`.

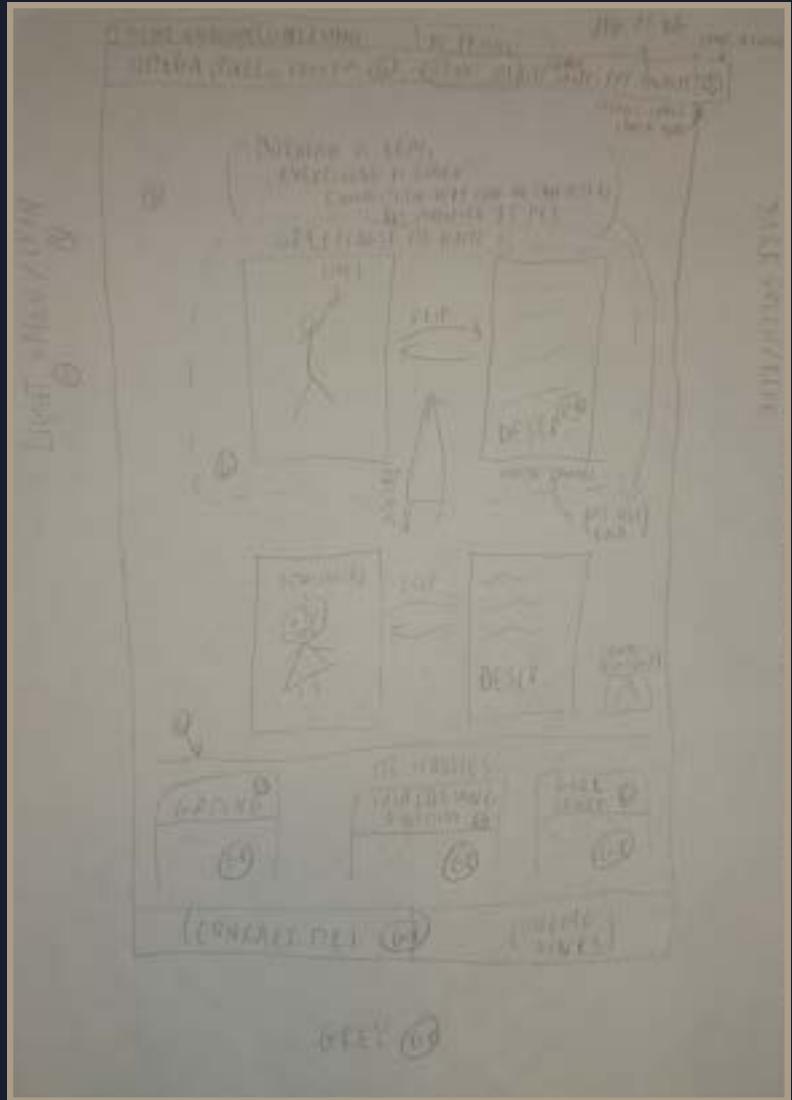
Successivamente, nella cartella del progetto, ho creato una sottocartella di nome assets, contenente all'interno altre tre cartelle: css, img e fonts.

Nella cartella css, ho salvato un file vuoto SCSS, che tramite l'estensione di *Watch Sass* di Visual Studio Code, creò automaticamente il file CSS.

## SUCCESSIVAMENTE, DEFINII DELLE BOZZE



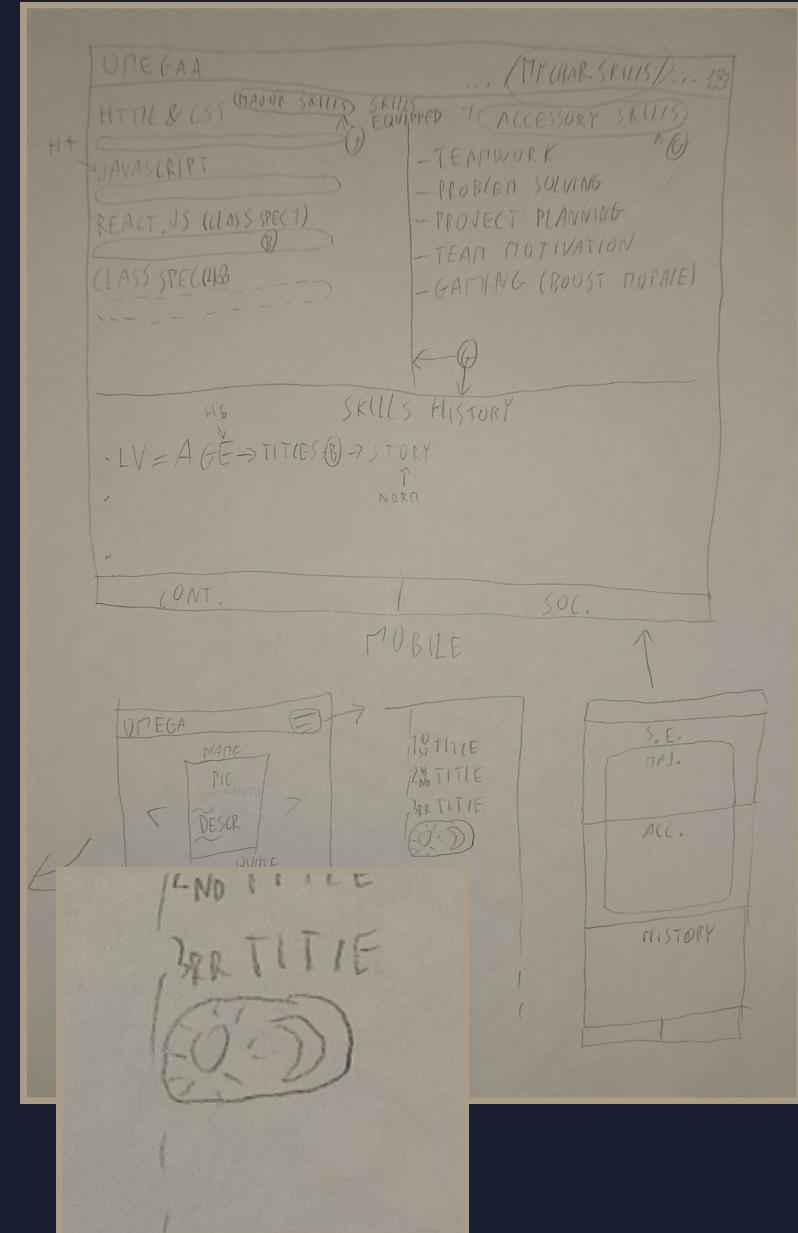
# BOZZE DEL PROGETTO



Ho immaginato una struttura basata su elementi nerd, colori sul blu e il verde, immagini scorrevoli e qualche chicca e citazione, seppur non tutto è finito nel progetto finale.

Il progetto avrebbe anche incluso un input per la modalità notturna, ma con una base codice SCSS piuttosto pesante che avrebbe comunque richiesto un minimo di codice JavaScript, non riuscii ad applicare con facilità il widget e decisi di abbandonare l'idea fino a prossimo aggiornamento.

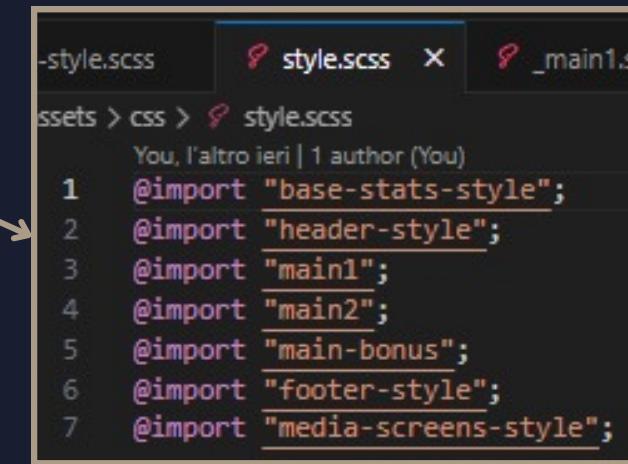
Concluso lo studio delle bozze, ho iniziato il progetto creando parziali SCSS, il quale chiamai uno di essi *\_base-stats*.



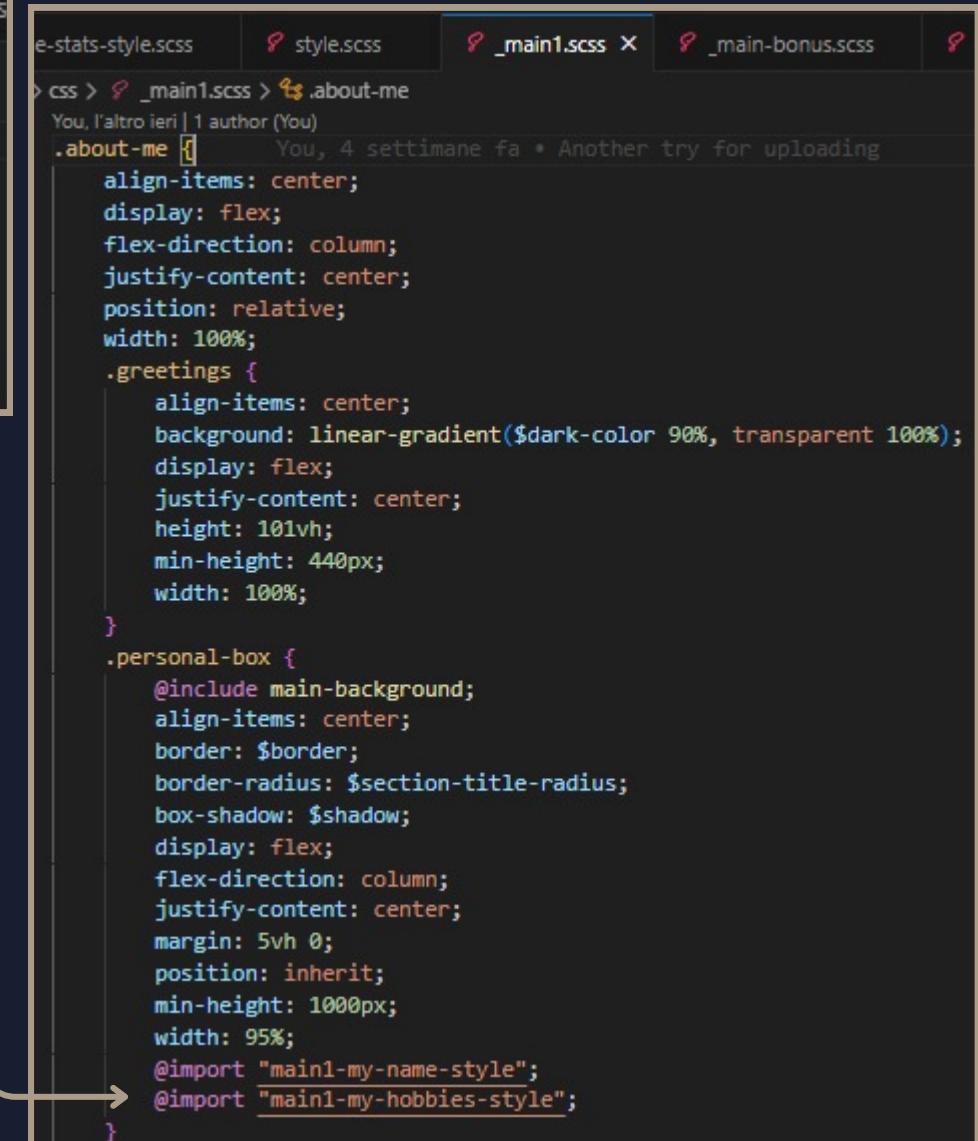
# PARTIALS/IMPORT

Ho creato vari file parziali (in principio `_base-stats-style`, `_header-style` e `_media-screens-style`, poi il resto) e che ho incluso all'interno di `style.scss` tramite `@import`.

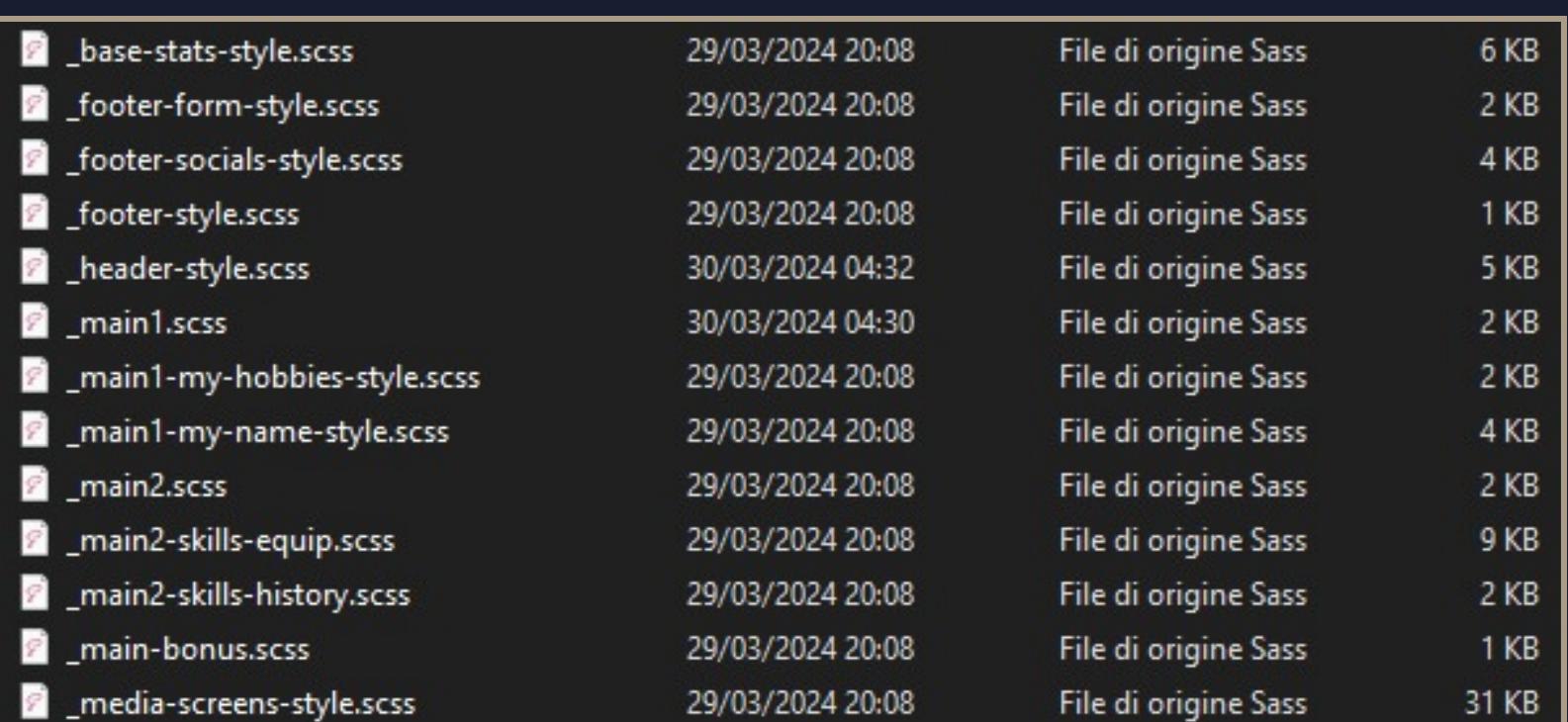
Vari altri parziali sono nati durante la progettazione e sono stati inclusi nei medesimi file SCSS *main* e *footer*...



```
-style.scss    style.scss    _main1.s
ssets > css > style.scss
You, l'altro ieri | 1 author (You)
1 @import "base-stats-style";
2 @import "header-style";
3 @import "main1";
4 @import "main2";
5 @import "main-bonus";
6 @import "footer-style";
7 @import "media-screens-style";
```



```
e-stats-style.scss    style.scss    _main1.scss    _main-bonus.scss
> css > _main1.scss > .about-me
You, l'altro ieri | 1 author (You)
.about-me [ You, 4 settimane fa • Another try for uploading
  align-items: center;
  display: flex;
  flex-direction: column;
  justify-content: center;
  position: relative;
  width: 100%;
.greetings {
  align-items: center;
  background: linear-gradient($dark-color 90%, transparent 100%);
  display: flex;
  justify-content: center;
  height: 101vh;
  min-height: 440px;
  width: 100%; }
.personal-box {
  @include main-background;
  align-items: center;
  border: $border;
  border-radius: $section-title-radius;
  box-shadow: $shadow;
  display: flex;
  flex-direction: column;
  justify-content: center;
  margin: 5vh 0;
  position: inherit;
  min-height: 1000px;
  width: 95%; }
  @import "main1-my-name-style";
  @import "main1-my-hobbies-style"; }
```



 _base-stats-style.scss	29/03/2024 20:08	File di origine Sass	6 KB
 _footer-form-style.scss	29/03/2024 20:08	File di origine Sass	2 KB
 _footer-socials-style.scss	29/03/2024 20:08	File di origine Sass	4 KB
 _footer-style.scss	29/03/2024 20:08	File di origine Sass	1 KB
 _header-style.scss	30/03/2024 04:32	File di origine Sass	5 KB
 _main1.scss	30/03/2024 04:30	File di origine Sass	2 KB
 _main1-my-hobbies-style.scss	29/03/2024 20:08	File di origine Sass	2 KB
 _main1-my-name-style.scss	29/03/2024 20:08	File di origine Sass	4 KB
 _main2.scss	29/03/2024 20:08	File di origine Sass	2 KB
 _main2-skills-equip.scss	29/03/2024 20:08	File di origine Sass	9 KB
 _main2-skills-history.scss	29/03/2024 20:08	File di origine Sass	2 KB
 _main-bonus.scss	29/03/2024 20:08	File di origine Sass	1 KB
 _media-screens-style.scss	29/03/2024 20:08	File di origine Sass	31 KB

# \_BASE-STATS.SCSS

```
@font-face {  
    font-family: "introrustg-base2line";  
    src: url(..../fonts/introrustg-base2line.woff) format("opentype"), url(..../fonts/introrustg_base2line_macroman/introrustg-base2line-webfont.woff) format("woff"), url(..../fonts/introrustg_base2line_macroman/introrustg-base2line-webfont.woff2) format("woff2");  
    font-weight: normal;  
    font-style: normal;  
}
```

```
//borders  
$border: 1px solid $title-color;  
$border-alt: 1px solid $title-color-alt;  
$border-transparent: 1px solid transparent;  
$border-lg: 2px double $title-color;  
$border-lg-alt: 2px solid $title-color-alt;  
$border-radius: 10px;  
$section-title-radius: 50px;  
$text-stroke: 0.5px $dark-color;  
$text-stroke-special: 0.5px $title-color-alt;
```

... Più nello specifico, ho impostato un parziale funzionante come guida per valori personalmente definiti e che avrei aggiornato continuamente, man mano che le idee per il sito si materializzassero e si ripetessero: [\\_base-stats.scss](#).

All'interno del codice creai valori per: "at-rules" come **@font-face** e **@mixin**, stili base per alcuni tag HTML, colori, bordi, gruppi e altezze font, ombre e altro, incluso un bordo-debug per controllare le dimensioni dei box sviluppati in fasi di testing

```
//-box limits test-  
$debug: 1px solid □black;
```

```
//colors  
$altered-text: □#999;  
$dark-color: □#111;  
$footer-mid-color: □#04091c;  
$inner-box: □#2e465f;  
$logo-p-color: □white;  
$primary-color: □#262d42;  
$primary-color-scrollbar: □#394362;  
$secondary-color: □#063258;  
$secondary-color-scrollbar: □#094d88;  
$skill-disabled-container: □#333;  
$soft-shadow: □#555;  
$title-color: □#ab9c87; You, 2 s  
$title-color-alt: □#423b26;  
$visited-links-perm: □#d0d0d0;
```

```
* {  
    -moz-box-sizing: border-box;  
    -webkit-box-sizing: border-box;  
    box-sizing: border-box;  
    margin: 0;  
    padding: 0;  
    scroll-behavior: smooth;  
    ::-webkit-scrollbar {  
        width: 0.5vw;  
    }  
    ::-webkit-scrollbar-track {  
        background: transparent;  
    }  
    ::-webkit-scrollbar-thumb {  
        background: $altered-text;  
        border-radius: calc($border-radius/2);  
        &:hover {  
            background: $title-color;  
        }  
    }  
}  
body {  
    @include body-background;  
    font-family: $font-body;  
}  
h1,  
h2,  
summary,  
legend,  
h3,  
h4,  
h5,  
h6 {  
    color: $title-color;  
    font-family: $font-title;  
    font-weight: normal;  
    text-align: center;  
    text-shadow: $shadow;  
}  
h1 {  
    color: $altered-text;  
    font-size: $h1;  
}
```

# HEADER (1/2)

L'idea era quella di dividere l'header in due sezioni: logo ed elementi ipertestuali.

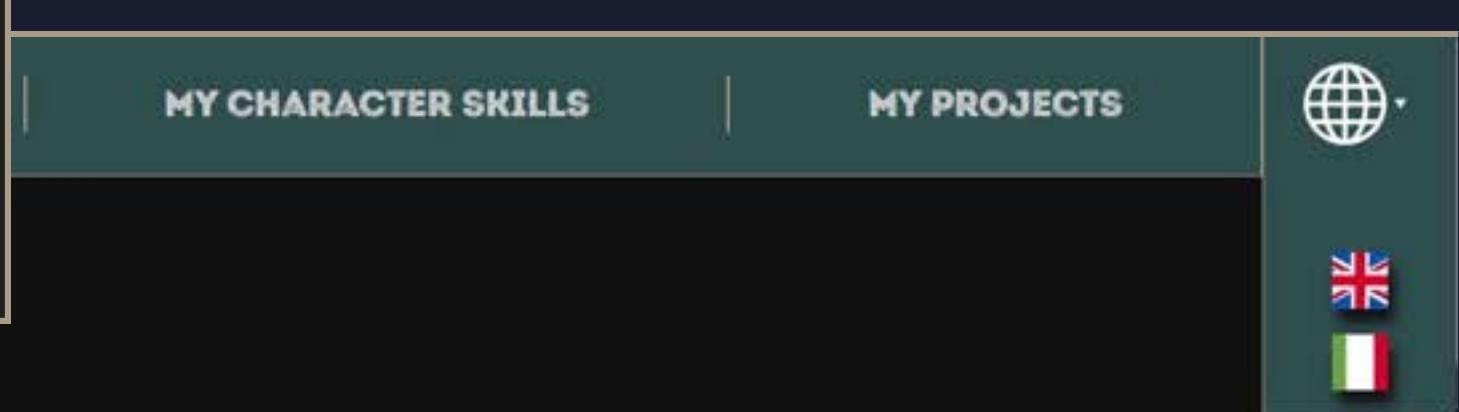
Per il logo, volevo utilizzare il volto preferito della mia favicon cercando di unirlo al mio tipico soprannome social, facendo funzionare il viso della mia figura come una "O" stilizzata. Tuttavia, l'immagine era molto più grande del suo contenitore, perciò, per ovviare al problema, sfruttai la proprietà `object-fit` ed il valore `contain`, nonché resi la larghezza dell'immagine adattabile in `7vmax` fino ad un minimo dell'equivalente di `56px`. Nel progetto, ho utilizzato spesso l'unità di misura in `viewport` per ovviare ad alcuni problemi come questo e rendere alcuni dettagli automaticamente adattabili (tipo imbottiture o margini dei contenitori). Anche l'altezza dell'header stesso.

```
<a href="index-it.html" name="logo" class="logo-container">
  <span class="logo-img">
    
    <em>MEGA</em>
  </span>
</a>
```

```
.logo-container,
.head-separator,
.head-section {
  display: flex;
}
.logo-container,
.head-section {
  position: inherit;
}
.head-separator,
.head-section {
  height: 100%;
}
.logo-container {
  justify-content: space-around;
  text-decoration: none;
}
.logo-img {
  display: flex;
}
.logo-alt {
  object-fit: contain;
  max-width: $sh1 * 2;
  width: 7vmax;
  min-width: $sh1;
}
em {
  align-self: center;
  color: $logo-p-color;
  display: block;
  font-size: $sh1;
```



```
header {
  background: linear-gradient(to bottom, $secondary-color 0%, $dark-color 100%);
  position: relative;
  max-height: $header-height-limits; (100px)
  height: 6vmax;
  min-height: $header-height-limits * 0.55;
  width: 100%;
```



# HEADER (1/2)

L'idea era quella di dividere l'header in due sezioni: logo ed elementi ipertestuali.

Per il logo, volevo utilizzare il volto preferito della mia favicon cercando di unirlo al mio tipico soprannome social, facendo funzionare il viso della mia figura come una "O" stilizzata. Tuttavia, l'immagine era molto più grande del suo contenitore, perciò, per ovviare al problema, sfruttai la proprietà `object-fit` ed il valore `contain`, nonché resi la larghezza dell'immagine adattabile in `7vmax` fino ad un minimo dell'equivalente di `56px`. Nel progetto, ho utilizzato spesso l'unità di misura in `viewport` per ovviare ad alcuni problemi come questo e rendere alcuni dettagli automaticamente adattabili (tipo imbottiture o margini dei contenitori). Anche l'altezza dell'header stesso.

Nella sezione degli *elementi ipertestuali*, divisi e separai gli oggetti con la proprietà `justify-content: space-evenly` e nascosi un input del menù a tendina con `display: none`, direzionando successivamente la mia attenzione verso un altro input per collegarlo ad un contenitore con posizione assoluta rispetto a quello con classe `.lang-stg` che avrebbe funto da menù per la scelta della lingua della pagina web. Ritornerò su questo caso più avanti nella presentazione.

```
<div class="head-menu">
  <input
    type="checkbox"
    name="menu-list"
    id="expandList" />
  <label
    for="expandList"
    class="head-menu-btn">
    
  </label>
</div>
```

```
.head-menu-btn {
  display: none;
  .menu-icon {
    display: none;
    filter: $inverted-filter;
  }
}
.head-menu-btn,
#expandList {
  display: none;
}
```



```
<div class="lang-stg">
  <input
    type="checkbox"
    name="language-menu"
    id="expandLang" />
  <label
    for="expandLang"
    class="lang-btn">
    
    <span>blacktriangledown</span>
  </label>
  <span class="lang-content">
    <a href="index.html"
      class="lang-item">
      
    </a>
    <a href="index-it.html"
      class="lang-item">
      
    </a>
  </span>
</div>
```

```
.lang-content {
  @include header-background;
  border: $border;
  border-right: none;
  border-top: none;
  border-bottom-right-radius: $border-radius;
  display: none;
  flex-direction: column;
  justify-content: flex-end;
  padding: 0 1vw 1vmax 1vw;
  position: absolute;
  z-index: 1;
  height: 14vmax;
  min-height: calc($min-viewport-width/2);
  left: 0;
  right: 0;
  top: 0;
}
.lang-item {
  display: flex;
  justify-content: flex-end;
  margin-top: 2vh;
  box-shadow: $shadow;
}
.lang-item-alt {
  width: 2vmax;
  min-width: $normal-font;
}
```



# HEADER (1/2)

L'idea era quella di dividere l'header in due sezioni: logo ed elementi ipertestuali.

---

Per il logo, volevo utilizzare il volto preferito della mia favicon cercando di unirlo al mio tipico soprannome social, facendo funzionare il viso della mia figura come una "O" stilizzata. Tuttavia, l'immagine era molto più grande del suo contenitore, perciò, per ovviare al problema, sfruttai la proprietà *object-fit* ed il valore *contain*, nonché resi la larghezza dell'immagine adattabile in 7vmax fino ad un minimo dell'equivalente di 56px. Nel progetto, ho utilizzato spesso l'unità di misura in *viewport* per ovviare ad alcuni problemi come questo e rendere alcuni dettagli automaticamente adattabili (tipo imbottiture o margini dei contenitori). Anche l'altezza dell'header stesso.

---

Nella sezione degli *elementi ipertestuali*, divisi e separai gli oggetti con la proprietà *justify-content: space-evenly* e nascosi un input del menù a tendina con *display: none*, direzionando successivamente la mia attenzione verso un altro input per collegarlo ad un contenitore con posizione assoluta rispetto a quello con classe *.lang-stg* che avrebbe funto da menù per la scelta della lingua della pagina web. Ritornerò su questo caso più avanti nella presentazione.

---

Come dissi in precedenza, nei miei piani era incluso anche un widget per la modalità notturna, trasformando il pulsante a forma di globo come una sezione impostazioni della pagina, ma mi complicò lo sviluppo abbastanza da dover ricominciare la creazione dell'header più volte per resettare l'ordine confuso dei codici, eventualmente abbandonando l'idea.



ABOUT ME

MY CHARACTER SKILLS

MY PROJECTS



# HEADER (1/2)

L'idea era quella di dividere l'header in due sezioni: logo ed elementi ipertestuali.

---

Per il logo, volevo utilizzare il volto preferito della mia favicon cercando di unirlo al mio tipico soprannome social, facendo funzionare il viso della mia figura come una "O" stilizzata. Tuttavia, l'immagine era molto più grande del suo contenitore, perciò, per ovviare al problema, sfruttai la proprietà *object-fit* ed il valore *contain*, nonché resi la larghezza dell'immagine adattabile in *7vmax* fino ad un minimo dell'equivalente di 56px. Nel progetto, ho utilizzato spesso l'unità di misura *viewport* per ovviare ad alcuni problemi come questo e rendere alcuni dettagli automaticamente adattabili (tipo imbottiture o margini dei contenitori). Anche l'altezza dell'header stesso.

---

Nella sezione degli *elementi ipertestuali*, divido e separo gli oggetti con la proprietà *justify-content: space-evenly* e nascondo un input del menù a tendina con *display: none*, direzionando successivamente la mia attenzione verso un altro input per collegarlo ad un contenitore con posizione assoluta rispetto a quello con classe *.lang-stg* che avrebbe funto da menù per la scelta della lingua della pagina web. Ritornerò su questo caso più avanti nella presentazione.

---

Come dissi in precedenza, nei miei piani era incluso anche un widget per la modalità notturna, trasformando il pulsante a forma di globo come una sezione impostazioni della pagina, ma mi complicò lo sviluppo abbastanza da dover ricominciare la creazione dell'header più volte per resettare l'ordine confuso dei codici, eventualmente abbandonando l'idea.

---

Infine, provai a decorare i vari link con le pseudo-classi *:hover*, *:active* e *:visited* e volevo definire le prime dimensioni per l'at-rule *@media screen*.



ABOUT ME

MY CHARACTER SKILLS

MY PROJECTS



# MEDIA SCREENS

Questa fase progettuale è stata incerta su come esporla per buona parte della costruzione della pagina web, avendo pochi riferimenti su cui basarmi per capire quali fossero i viewport ideali e come organizzarli al meglio. Provai, quindi, i seguenti metodi:

1. Guardando i separatori nell'header (ed i limiti dei contenitori nei successivi lavori), ho definito un breakpoint uno o poco più pixel prima che il testo andasse a capo automaticamente, sovrapponendosi ai contenitori, ma gli stili dei font e delle immagini del tag `<main>` non permetteva un buon funzionamento e rispetto dei limiti interni;
2. Controllare viewport esistenti di vari dispositivi mobile e laptop (sia in verticale che orizzontale) registrati nel DevTool di *Google Chrome* e creare media screens appositi per ognuno di essi, eventualmente unendo in eccesso larghezze molto simili. Definii, inoltre, dei media screens anche per le altezze massime con lo stesso metodo.

L'ultimo metodo ha funzionato per un po' di tempo, ma alcune dimensioni, seppur nei limiti dei contenitori, risultavano sproporzionati, mentre i media screens per le altezze massime cominciavano ad apparire superflui a causa dell'unità di misura viewport, quindi li encapsurai come commenti per un po', poi eliminati definitivamente.

# MEDIA SCREENS

Vidi questa progettazione da un altro punto di vista, regolandomi con dei dati particolari:

- Ho presupposto che la risoluzione dello schermo fosse quella base da cui partire: 1920px;
- Lo schermo più stretto registrato nel *Google Chrome DevTool* è quello del *Samsung Galaxy Fold*: 280px;
- *Visual Studio Code* ha una dimensione font preimpostata nella maggior parte dei tag che ho inteso come minimo: 16px;
- Per rendere più uniformi le proporzioni, ho immaginato un numero costante che diminuisse di 0.05 punti ad ogni viewport, fino ad un minimo di 0.5 (*Galaxy Fold*) ed un massimo di 1 (1920px).

Da questi dati, ho distribuito dieci diversi viewport e in ognuno di essi ho calcolato le dimensioni di varie proprietà con le costanti appena descritte. Unica regola era quella di non diminuire certe dimensioni oltre il limite minimo di 16px, tenendo quindi fuori dai calcoli alcune proprietà nei viewport più piccoli.

```
69 /*SPECIAL - Screens > 1920px*/
70 > @media screen and (min-width: 1921px) {
487 }
488
489 > @media screen and (max-width: 1756px) {
523 }
524
525 > @media screen and (max-width: 1592px) {
639 }
640
641 > @media screen and (max-width: 1428px) {
755 }
756
757 > @media screen and (max-width: 1264px) {
871 }
872
873 > @media screen and (max-width: 1100px) {
958 }
959
960 > @media screen and (max-width: 936px) {
1045 }
1046
1047 > @media screen and (max-width: 772px) {
1523 }
1524
1525 > @media screen and (max-width: 608px) {
1596 }
1597
1598 > @media screen and (max-width: 444px) {
1673 }
1674
1675 /*GALAXY FOLD - Screens <= 280px*/
1676 > @media screen and (max-width: 280px) {
1728 }
```

```
/*GALAXY FOLD - Screens <= 280px*/
@media screen and (max-width: 280px) {
    /*Constant: 0.5=calc(x/2) - no $form-font, $normal-font, $h5, $h4, $h3*/
```

# RICERCA DEGLI SFONDI



Sito per definire i colori principali degli sfondi e loro combinazioni



DEEP AI

Sito di immagini generate tramite IA da cui si basano sfondo principale (stanza) e di testo (come questo)



Sito di origine di alcune foto e lo sfondo per i titoli (ad esempio, quello del logo)

Molto presto nello sviluppo, notai che l'uso dei colori da me scelti davano un'idea estetica desueta, per cui mi basai su questi tre siti per cercare immagini libere da copyright e basandomi su un carattere più "lo-fi". Inoltre, la combinazione blu/verde della mia bozza divenne argilla ebano/rosa caverna.

# HEADER (2/2)

Continuando lo sviluppo dell'header da *@media screen and (max-width: 772px)*, riattivo l'icona del menù dei link dell'header tramite *display: block*, ma lo sposto anche al lato destro del pulsante delle lingue, stilizzandolo con la pseudo-classe *:active*.

Inoltre, ho cambiato la posizione in assoluto della lista dei link, aggiungendo anche la proprietà *overflow: hidden* per nascondere l'altezza del suo container sotto l'header.

```
.head-section {  
    flex-direction: row-reverse;  
    justify-content: flex-start;  
    .head-menu {  
        align-items: flex-end;  
        border: $border;  
        border-radius: $border-radius;  
        display: flex;  
        flex-direction: column;  
        margin: 0 1vw;  
        position: relative;  
        width: auto;  
        .head-menu-btn,  
        .head-list {  
            display: flex;  
        }  
        .head-menu-btn {  
            cursor: pointer;  
            border-radius: $border-radius;  
            z-index: 2;  
            .menu-icon {  
                display: block;  
                width: 4vmax;  
                min-width: calc($h1/2); //FIXED  
            }  
            &:active {  
                background-color: transparentize($dark-color, 0.5);  
            }  
        }  
        #expandlist {  
            display: none;  
            &:checked {  
                .head-list {  
                    display: flex;  
                    height: 75px;  
                }  
            }  
        }  
        .head-list {  
            @include base-transition;  
            align-items: flex-end;  
            flex-direction: column;  
            justify-content: flex-end;  
            overflow: hidden;  
            position: absolute;  
            z-index: 1;  
            height: 0;  
            width: $min-viewport-width;  
            right: -5px;  
            top: 45px;  
        }  
    }  
}
```

Per spostare  
l'icona menù  
dei link



# MAIN.ABOUT-ME

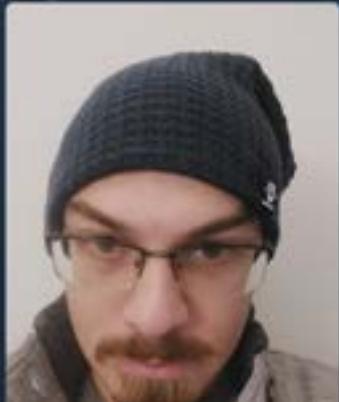
**GREETINGS!  
I SHALL INTRODUCE MYSELF.  
MY NAME IS...**

La mia intenzione era quella di creare un lento inizio per la mia presentazione con un semplice quadro di benvenuto. Lo stile era un contenitore a sfondo scuro che ricopriva l'intero viewport in altezza e poco oltre per motivi che presenterò in seguito (*height: 101vh*) con un testo sviluppato sul tag *<h1>*.

# MAIN.ABOUT-ME

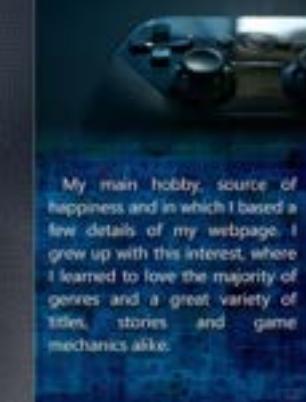
## EMANUELE ROMANO (AKA OMEGA)

I'm a Full Stack Development student at **Start2Impact University**. Feel free to poke around my website, widgets and links! For starters, click these pictures to show a few more details.



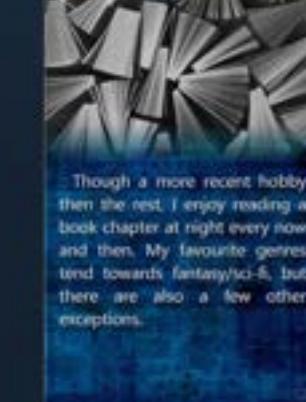
## MY HOBBIES

### GAMING



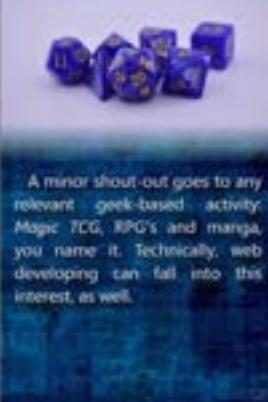
My main hobby, source of happiness and in which I based a few details of my webpage. I grew up with this interest, where I learned to love the majority of genres and a great variety of titles, stories and game mechanics alike.

### READING



Though a more recent hobby than the rest, I enjoy reading a book chapter at night every now and then. My favourite genres tend towards fantasy/sci-fi, but there are also a few other exceptions.

### GEEK STUFF



A minor shout-out goes to any relevant geek-based activity: Magic TCG, RPG's and manga, you name it. Technically, web developing can fall into this interest, as well.

Oltre la prima presentazione, ho creato un contenitore dal nome *.personal-box* e in cui ho inserito due sezioni: *.my-name* e *.my-hobbies*. Entrambi dovevano contenere delle immagini su di me e sui miei passatempi, appunto, con delle brevi descrizioni personali.

In entrambi i casi, a seconda delle dimensioni del viewport, le immagini dovevano fungere come slide scorrevoli, ma le avrei programmate successivamente; rispetto alla sezione *.my-hobbies*, nel quale le descrizioni erano subito visibili con un effetto simil-trasparente con *mask-image: linear-gradient(\$secondary-color 80%, transparent)*, in *.my-name* avrei dovuto sfruttare l'effetto degli overlay.

# TESTO IN OVERLAY

Per il testo in overlay, dovevo sviluppare una box a parte, comunque inserita all'interno di `.name-slide-content`, che avrei codificato con posizione assoluta e fissato alla base dell'immagine, la quale era inserita all'interno del tag `<label>`. Chiaramente, questi contenitori erano tutti seguiti dal tag `<input type="checkbox">`, il quale nascosi tramite `display: none`. A seguire, grazie alla pseudo-classe `:checked`, collegai gli stati che doveva seguire l'altezza dell'overlay a seconda che l'input fosse selezionato o meno, per poi aggiungere `overflow-y: scroll` per ovviare ai possibili problemi che mi avrebbero causato le differenze di viewport, font e altezza delle immagini.

```
<div class="name-slide">
  <picture class="name-slide-content">
    <input
      type="checkbox"
      name="image-description"
      id="nameOverlayCheck1" />
    <label
      for="nameOverlayCheck1"
      class="name-overlay-btn">
      
    </label>
    <span class="name-overlay">
      <p class="name-img-desc">This is me, trying to look good on those rare times I make pictures of myself.</p>
    </span>
  </picture>
</div>
```

```
#nameOverlayCheck1,
#nameOverlayCheck2 {
  display: none;
}

#nameOverlayCheck1 {
  &:checked {
    + .name-overlay {
      height: 23%;
    }
  }
}

#nameOverlayCheck2 {
  &:checked {
    + .name-overlay {
      height: 28%;
    }
  }
}

.name-overlay {
  @include base-transition;
  align-items: flex-start;
  background-color: transparentize($dark-color, 0.3);
  border-bottom-left-radius: $border-radius;
  border-bottom-right-radius: $border-radius;
  overflow-y: scroll;
  position: absolute;
  height: 0;
  bottom: 0;
  .name-img-desc {
    display: block;
    padding: 1vh 1vw;
    position: relative;
    text-align: justify;
    text-indent: 0.8em;
  }
}
```



This is me, trying to look good on those rare times I make pictures of myself.

# IMMAGINI SCORREVOLI

```
.name-cards {
    align-items: flex-start;
    overflow: hidden;
    &:not(:last-child) {
        &:hover {
            .name-bullets {
                .name-bullet-effect {
                    &:first-of-type {
                        visibility: visible;
                    }
                }
            }
        }
    }
}

&:checked {
    &:not(:last-child) {
        &:hover {
            .name-bullets {
                .name-bullet-effect {
                    &:first-of-type {
                        visibility: visible;
                    }
                }
            }
        }
    }
}

&:not(:checked) {
    &:not(:last-child) {
        &:hover {
            .name-bullets {
                .name-bullet-effect {
                    &:first-of-type {
                        visibility: visible;
                    }
                }
            }
        }
    }
}
```

```
.name-cards {
    align-items: center;
    align-self: center;
    display: flex;
    justify-content: space-evenly;
    margin: 2vh 0;
    width: 25%;
}

.name-bullet-effect {
    color: $title-color;
    display: block;
    font-size: $h3 * 0.65;
    visibility: hidden;
}

.name-bullet-points {
    -moz-transition: background-color 0.5s ease-out;
    -o-transition: background-color 0.5s ease-out;
    -webkit-transition: background-color 0.5s ease-out;
    background-color: #ccc;
    border: 2px solid transparent;
    border-radius: 100px;
    box-shadow: $shadow;
    cursor: pointer;
    margin: 0 0.5em;
    opacity: 0.5;
    transition: background-color 0.5s ease-out;
    height: 4vmin;
    min-height: 15px;
    width: 4vmin;
    min-width: 15px;
    &:hover {
        -moz-transition: background-color 0.5s ease-in;
        -o-transition: background-color 0.5s ease-in;
        -webkit-transition: background-color 0.5s ease-in;
        background-color: black;
        transition: background-color 0.5s ease-in;
    }
}
```

Sempre in un viewport a 772px di larghezza, ho creato una sequenza di input radio di due elementi che chiamai `.name-bullets` e ho fatto in modo che il contenitore di entrambe le immagini (`.name-inner`) fosse largo il doppio di come era impostato, ossia del 200%. Dopodiché, ad ogni ID delle slide, registravo un margine sinistro di 0 per la prima foto, -100% per la seconda.

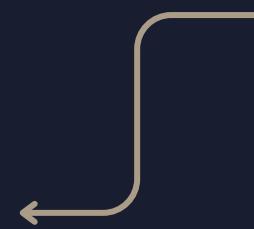


# IMMAGINI SCORREVOLI

```
.name-cards {
    align-items: flex-start;
    overflow: hidden;
}
.name-slide {
    &:hover {
        .name-bullets {
            .name-bullet-effect {
                &:last-of-type {
                    visibility: visible;
                }
            }
        }
    }
    &:checked {
        .name-inner {
            margin-left: 0;
        }
        .name-bullets {
            .name-bullet-points {
                &:nth-child(2) {
                    background-color: $secondary-color;
                    background-image: url('../img/background-title.jpg');
                }
            }
        }
    }
}
.name-slide2 {
    &:hover {
        .name-bullets {
            .name-bullet-effect {
                &:last-of-type {
                    visibility: visible;
                }
            }
        }
    }
    &:checked {
        .name-inner {
            margin-left: -100px;
        }
        .name-bullets {
            .name-bullet-points {
                &:nth-child(4) {
                    background-color: $secondary-color;
                    background-image: url('../img/background-title.jpg');
                }
            }
        }
    }
}
```

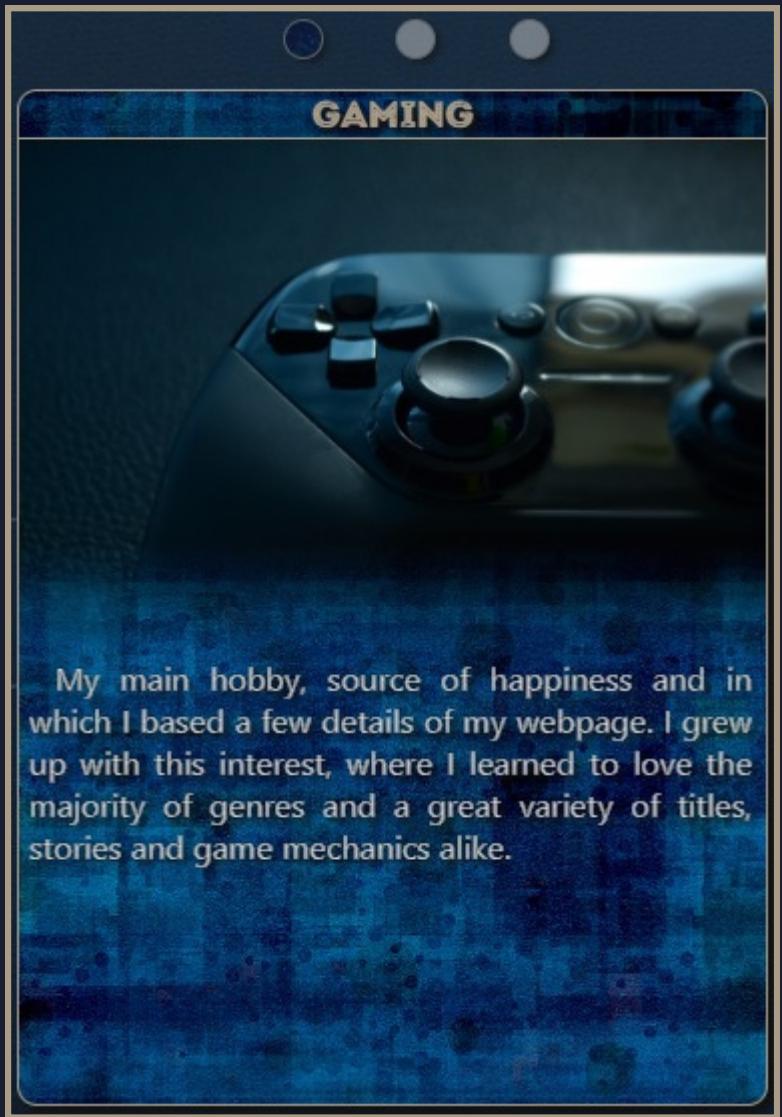
```
.name-cards {
    align-items: center;
    align-self: center;
    display: flex;
    justify-content: space-evenly;
    margin: 2vh 0;
    width: 25%;
}
.name-bullet-effect {
    color: $title-color;
    display: block;
    font-size: $h3 * 0.65;
    visibility: hidden;
}
.name-bullet-points {
    -moz-transition: background-color 0.5s ease-out;
    -o-transition: background-color 0.5s ease-out;
    -webkit-transition: background-color 0.5s ease-out;
    background-color: #ccc;
    border: 2px solid;
    border-radius: 100px;
    box-shadow: $shadow;
    cursor: pointer;
    margin: 0 0.5em;
    opacity: 0.5;
    transition: background-color 0.5s ease-out;
    height: 4vmin;
    min-height: 15px;
    width: 4vmin;
    min-width: 15px;
}
&:hover {
    -moz-transition: background-color 0.5s ease-in;
    -o-transition: background-color 0.5s ease-in;
    -webkit-transition: background-color 0.5s ease-in;
    background-color: black;
    transition: background-color 0.5s ease-in;
}
```

Sempre in un viewport a 772px di larghezza, ho creato una sequenza di input radio di due elementi che chiamai `.name-bullets` e ho fatto in modo che il contenitore di entrambe le immagini (`.name-inner`) fosse largo il doppio di come era impostato, ossia del 200%. Dopodiché, ad ogni ID delle slide, registravo un margine sinistro di 0 per la prima foto, -100% per la seconda.



Come aggiunta, ho incluso anche degli `<span>` con un segnale a triangolo che sarebbe comparso tramite pseudo-classe `:hover` e proprietà `visibility: visible` alla sinistra del bullet, come se fosse un selettori di un videogioco di ruolo.

# IMMAGINI SCORREVOLI



Nella sezione *.my-hobbies*, come unica differenza al numero maggiore di immagini usate, ampliai la larghezza di *.hobby-inner* a 300% e inclusi un ID in più con un margine sinistro di -200%.

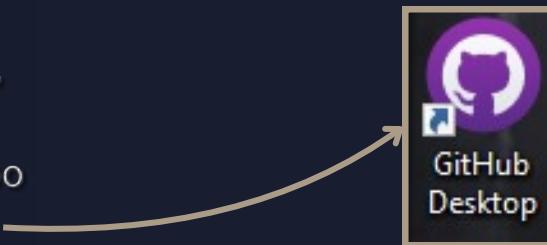
# GITHUB

L'approccio con **GitHub** avvenne in ritardo per l'unica ragione di non essere stato (e forse nemmeno tuttora) pratico del portale, nonostante il tutorial del corso, ma eventualmente mi decisi ad importare il progetto anche lì.

Avevo collegato, innanzitutto, un mio vecchio profilo personale con *Visual Studio Code* che utilizzavo solo per scopi di raccolta di mod per videogiochi. Inoltre, sempre dall'editor di codici, installai quasi tutte le estensioni *Git* ufficiali e cominciai a fare dei tentativi di caricamento di un lavoro tutorial esterno al progetto.

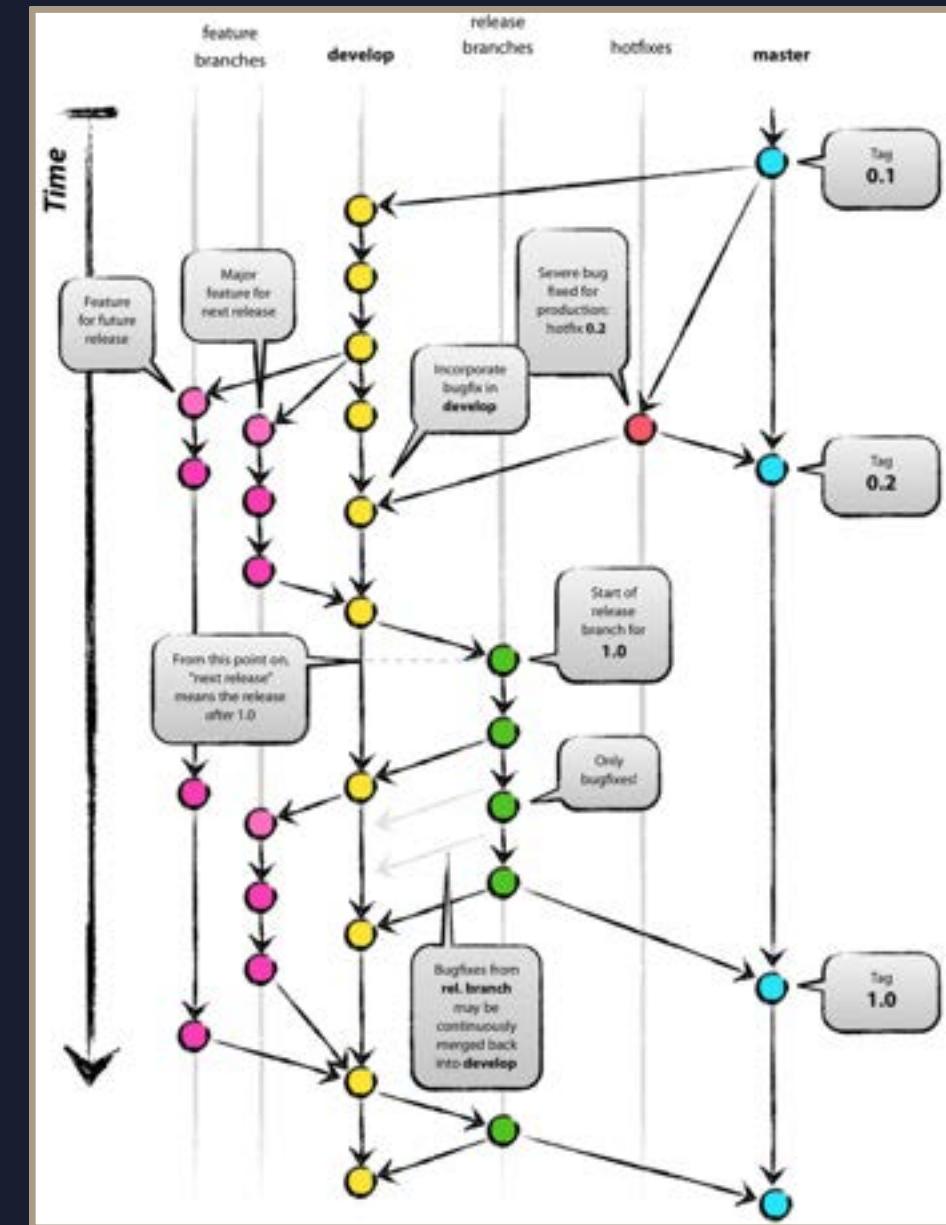
Tutto funzionò senza intoppi, ma decisi di cambiare idea col mio profilo e ne aggiunsi uno nuovo collegato alla mia email professionale, cominciando ad avere qualche problema di riconoscimento profili e facendo continui tentativi di caricamento del mio progetto online.

Comunque, riuscii a comprendere le impostazioni del codice sorgente di *Visual Studio Code* e le varie azioni di creazione branch, commit, push e pull requests. Inoltre, notai anche l'importanza di *GitHub Desktop* come applicazione di supporto ed eventualmente, giorno dopo giorno, registrai nuovi commit nel mio profilo.



# GITHUB

Col senso di poi, una delle cose che terrò meglio in considerazione sarà quella di creare più branch al di fuori di *develop* e *main* per organizzare meglio lo storico dello sviluppo, seppur sento di dover imparare ancora molto da questo portale.



# SCROLLBAR SOTTILE

```
::-webkit-scrollbar {  
    width: 0.5vw;  
}  
  
::-webkit-scrollbar-track {  
    background: transparent;  
}  
  
::-webkit-scrollbar-thumb {  
    background: $altered-text;  
    border-radius: calc($border-radius/2);  
    &:hover {  
        background: $title-color;  
    }  
}
```

Per scelta personale, desideravo una forma dello scrollbar più sottile della normale forma dei browser. Per risolvere il problema, ho usato le pseudo-classi `::-webkit` specifiche desiderate per le parti dello scrollbar che volevo cambiare.



# FOOTER

Il corpo del footer è stato relativamente semplice da realizzare. Prima di tutto, creai due sezioni: contatti e collegamenti, per cui a loro volta definii una larghezza del 50% ognuno.

La sezione collegamenti era definita da `<ul>` e `<i>` vari che stilizzai in un altro momento con degli effetti simili ai bullet delle immagini scorrevoli e seguendo una filosofia simile.

Nella sezione contatti, definii vari `<fieldset>`, `<input>` e `<label>` che mi sarebbero stati utili per l'impostazione di EmailJS.

Si ringrazia, inoltre, **Alessandro Tozzi** e **Start2Impact** stessa per avermi reso disponibile l'immagine del loro logo in formato PNG

**CONTACT ME!**

**FULL NAME \***  
Basch Von Ronseburg

**EMAIL \***  
example@email.com

**YOUR MESSAGE \***  
Type your message here...

Submit    Reset



**CONTACT ME!**

**FULL NAME \***  
Basch Von Ronseburg

**EMAIL \***  
example@email.com

**YOUR MESSAGE \***  
Type your message here...

Submit    Reset



# EMAILJS

```
<script type="text/javascript" src="https://cdn.jsdelivr.net/npm/@emailjs/browser@4/dist/email.min.js"></script>
<script type="text/javascript">
const contactForm = document.getElementById("contact_form");
const serviceID = "service_y7neroj";
const templateID = "template_ygmb70h";
var submitBtn = document.getElementById("submitBtn");
var userName = document.getElementById("user_name");
var userEmail = document.getElementById("user_email");
var userMessage = document.getElementById("user_message");

(function () {
  emailjs.init({
    publicKey: "GK1P-15e5eEyCffxK",
  });
})();
window.onload = function () {
  contactForm.addEventListener("submit", function (event) {
    event.preventDefault();
    submitBtn.value = "Pending...";
    emailjs.sendForm(serviceID, templateID, this).then(
      () => {
        console.log("SUCCESS!");
        alert("Email sent!");
        submitBtn.value = "Submit";
        userName.value = "";
        userEmail.value = "";
        userMessage.value = "";
      },
      (error) => {
        console.log("FAILED...", error);
        alert(JSON.stringify(error));
      }
    );
  });
};
```

La guida per la costruzione di un servizio email esterno è stato facile da seguire, se non addirittura divertente nella possibilità che gli ID del form venissero riconosciuti per essere applicati nei testi elettronici, ma trovai la presentazione dello script come se certe sue variabili fossero in qualche modo prese per scontate. Fra queste, trovai problemi coi seguenti elementi:

- L'azione del tag `<form>`, che nonostante venisse creata automaticamente, sembrava essere controproducente per il funzionamento del sito intero;
- L'ID `contact_number`, che nell'esempio del sito veniva presentato come un numero statico senza spiegare l'applicazione di un contatore automatico. Eventualmente scartai l'idea perché, oltre che essere più complesso delle mie conoscenze, risultava vagamente "industriale";
- La mancanza di un segnale visivo del sito che avrebbe mostrato all'utente l'invio della sua email e non solo una registrazione nascosta nel DevTool del successo dell'azione svolta;
- La divisione degli script in più tag e le loro posizioni nel file HTML

# EMAILJS

```
<script
  type="text/javascript"
  src="https://cdn.jsdelivr.net/npm/@emailjs/browser@4/dist/email.min.js"></script>
<script type="text/javascript">
  const contactForm = document.getElementById("contact_form");
  const serviceID = "service_y7neroj";
  const templateID = "template_ygmb70h";
  var submitBtn = document.getElementById("submitBtn");
  var userName = document.getElementById("user_name");
  var userEmail = document.getElementById("user_email");
  var userMessage = document.getElementById("user_message");

  (function () {
    emailjs.init({
      publicKey: "GK1P-15e5eEyCffxK",
    });
  })();
  window.onload = function () {
    contactForm.addEventListener("submit", function (event) {
      event.preventDefault();
      submitBtn.value = "Sending...";
      emailjs.sendForm(serviceID, templateID, this).then(
        () => {
          console.log("SUCCESS!");
          alert("Email sent!");
          submitBtn.value = "Submit";
          userName.value = "";
          userEmail.value = "";
          userMessage.value = "";
        },
        (error) => {
          console.log("FAILED...", error);
          alert(JSON.stringify(error));
        }
      );
    });
  });
}; You, 2 settimane fa via PR #2 • - Changed "about-me", "my-name" and "my-h
```

Eppure, riuscii a completare anche questo obiettivo e resi il mio form funzionante al 100%, applicando solo qualche tocco personale che ho visto durante i miei studi di Bootstrap (come i *const* e *var*) e facendo funzionare il resto grazie all'aiuto della **Referente Sviluppo Patrizia**.

# FILTRI COLORI SOCIAL

Nato come un esperimento nella mia idea e messi in fila i file SVG dei link social, volevo trovare un metodo per colorare le immagini all'interno del codice, trovando un sito internet interessante di **CodePen** con uno script funzionante all'occasione che si basava sull'unione di vari filtri per raggiungere i colori desiderati. Il sito esatto è: <https://codepen.io/sosuke/pen/Pjoqqp>

Target color #00a4d6

Compute Filters

Real pixel, color applied through CSS background-color:



Filtered pixel, color applied through CSS filter:



The goal was to be able to create custom style sheets and allow for the coloring of icons for [Creating a Dovetail Agent Theme](#).

For this code to work well the starting color needs to be black. If your icon set isn't black you can prepend "brightness(0) saturate(100%)" to your filter property which will first turn the icon set to black.

For as long as I worked on creating this solution from multiple resources I found some had spent far longer to create this already completed solution. Only slightly modified to focus on HEX colors. Credit goes to [MultiplyByZero](#) for their post <https://stackoverflow.com/a/43960991/604861>

```
//filters
$facebook-filter: invert(31%) sepia(100%) saturate(325%) hue-rotate(181deg) brightness(90%) contrast(90%);
$github-filter: invert(29%) sepia(7%) saturate(683%) hue-rotate(178deg) brightness(94%) contrast(97%);
$instagram-filter: invert(63%) sepia(13%) saturate(1550%) hue-rotate(316deg) brightness(107%) contrast(90%);
$inverted-filter: invert(100%); //Also X filter
$linkedin-filter: invert(23%) sepia(34%) saturate(4379%) hue-rotate(179deg) brightness(114%) contrast(97%);
$threads-filter: invert(0%) sepia(17%) saturate(1102%) hue-rotate(353deg) brightness(92%) contrast(87%);
$title-filter: invert(62%) sepia(12%) saturate(549%) hue-rotate(356deg) brightness(101%) contrast(83%); //Like $title-color
```



```
span {
  align-items: center;
  display: flex;
  flex-direction: column-reverse;
  justify-content: flex-start;
  height: 100%;
  &:first-of-type {
    a {
      &:hover {
        filter: $facebook-filter;
      }
    }
    &:nth-of-type(2) {
      a {
        &:hover {
          filter: $instagram-filter;
        }
      }
    }
    &:nth-of-type(3) {
      a {
        &:hover {
          filter: $threads-filter;
        }
      }
    }
    &:nth-of-type(4) {
      a {
        &:hover {
          filter: $inverted-filter;
        }
      }
    }
    &:last-of-type {
      a {
        &:hover {
          filter: $linkedin-filter;
        }
      }
    }
  }
}
```

# PULSANTE BACK-TO-TOP

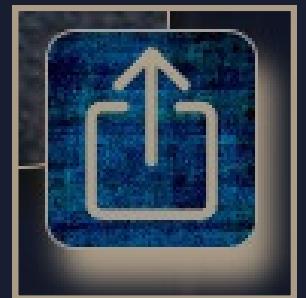
```
.max-sticky-height {  
    display: flex;  
    position: absolute;  
    height: 70%;  
    bottom: 1vh;  
    right: 0;  
    a {  
        @include base-transition;  
        @include title-background;  
        align-self: flex-end;  
        border: $border;  
        border-radius: $border-radius;  
        box-shadow: $shadow;  
        margin: 0 1vw 1vh 0;  
        padding: 5px 5px;  
        position: sticky;  
        min-width: 35px;  
        bottom: 1vh;  
        img {  
            filter: $title-filter;  
            width: 3vw;  
            min-width: 35px;  
        }  
        &:hover {  
            box-shadow: $shadow-special;  
        }  
    }  
}
```

```
<div class="max-sticky-height">  
    <a href="#logo">  
          
    </a>  
</div>
```

Presupponendo che fosse faticoso per l'utente che visita la mia pagina web ritornare all'header scrollando la pagina fino alla parte superiore, ho voluto includere nelle classi *main* un pulsante *back-to-top* collegato al logo. Ma al click, avrebbe trasportato l'utente all'istante senza alcuna animazione.

Per risolvere questo problema minore, ho applicato a *\** la semplice proprietà *scroll-behavior: smooth*.

```
* {  
    -moz-box-sizing: border-box;  
    -webkit-box-sizing: border-box;  
    box-sizing: border-box;  
    margin: 0;  
    padding: 0;  
    scroll-behavior: smooth;
```



# KEYFRAME & ANIMAZIONI

```
@media (prefers-reduced-motion: no-preference) {  
  @keyframes fade-in-box {  
    from {  
      scale: 0.5;  
      opacity: 0;  
    }  
    to {  
      scale: 1;  
      opacity: 1;  
    }  
  }  
  @keyframes fade-in {  
    from {  
      opacity: 0;  
    }  
    to {  
      opacity: 1;  
    }  
  }  
  @keyframes fade-out {  
    40% {  
      opacity: 1;  
    }  
    85%,  
    100% {  
      opacity: 0;  
    }  
  }  
}
```

Come sostituto ad una presumibile animazione JavaScript, cercai informazioni su internet e scoprii il funzionamento dell'at-rule **@keyframes**.

Evitai di creare animazioni troppo complesse, per il momento, ma applicai qualcosa che potesse apparire interessante alla prima occhiata della pagina index, creando animazioni per le classi `.greetings` (che resi più alta apposta per un migliore offset di animazione), `.personal-box`, `.my-name` e `.my-hobbies` e includendo la proprietà `prefers-reduced-motion: no-preference` a `@media` come soluzione anti cinetosi.

```
.personal-box > .my-hobbies {  
  animation: fade-in linear;  
  animation-range-start: cover;  
  animation-range-end: 20vh;  
  animation-timeline: view();  
}
```

# KEYFRAME & ANIMAZIONI

## NOTA SULLE COMPATIBILITÀ BROWSER

Purtroppo, vi è da considerare che la proprietà *animation* di CSS è tuttora considerata sperimentale e non funzionante su tutti i browser, con maggiore attenzione verso *Mozilla Firefox*, nel quale leggeva l'animazione della classe *.greetings* in maniera diversa da quella presupposta, quindi come soluzione temporanea ho incluso l'at-rule **@support** nell'omonima classe ed uno script di internet nei file HTML che avrebbe disattivato queste proprietà completamente nel browser.

```
/*FireFox locker due to incompatibility of the browser*/
@supports (animation-timeline: view()) {
  .about-me > .greetings {
    animation: fade-out linear forwards;
    animation-range-start: 25vh;
    animation-range-end: cover;
    animation-timeline: view();
  }
}
```

```
<!--ANIMATION SCRIPT FOR FIREFOX COMPATIBILITY-->
<script src="https://flackr.github.io/scroll-timeline/dist/scroll-timeline.js"></script>
```



Firefox OS

# MAIN.MY-SKILLS

## MY CHARACTER SKILLS

MAJOR SKILLS	ACCESSORY SKILLS
<b>HTML &amp; CSS</b>	<b>TEAMWORK</b> Adaptability to work in a team <i>+1 to Perception Check</i>
<b>JAVASCRIPT</b> <b>(ONGOING)</b> *	<b>PROBLEM ANALYSIS</b> General understanding of issues to reach the best solution suited for the case and time <i>+1 to Intelligence Check</i>
<small>* Certain functions required to know a few scripts to work, like Bootstrap or Emails</small>	<b>PROJECT PLANNING</b> Ease project focus by dividing the work in multiple breakpoints <i>+1 Tactics Slot</i>
<b>CLASS SPECIALIZATION #1: REACT.JS</b>	<b>TEAM MOTIVATION</b> Boosts others morale <i>+1 to Charisma Check</i>
<b>CLASS SPECIALIZATION #2: N/A</b>	<b>GAMING</b> Boost own morale <i>+1% to Major/Minor Skills</i>
<b>▶ MINOR SKILLS:</b>	(More of my skills in my <a href="#">LinkedIn profile page</a> )

**▶ SKILLS HISTORY**

Come immaginato anche dalla mia bozza, la seconda pagina dovrebbe raffigurare un'unione stilistica fra le statistiche di completamento delle abilità di un videogioco GDR (sfruttando codici CSS base di posizioni relative ed assolute per fare gli effetti estetici bonus nelle barre) nelle *Major Skills*, sia delle caratteristiche di costruzione personaggio così come appaiono in un manuale di un gioco da tavolo nelle *Accessory Skills*. Da qui, il titolo personale di **My Character Skills**.

# MAIN.MY-SKILLS

## ▼ SKILLS HISTORY

**GAMING**

Context: Personal  
Date: ~ 2000

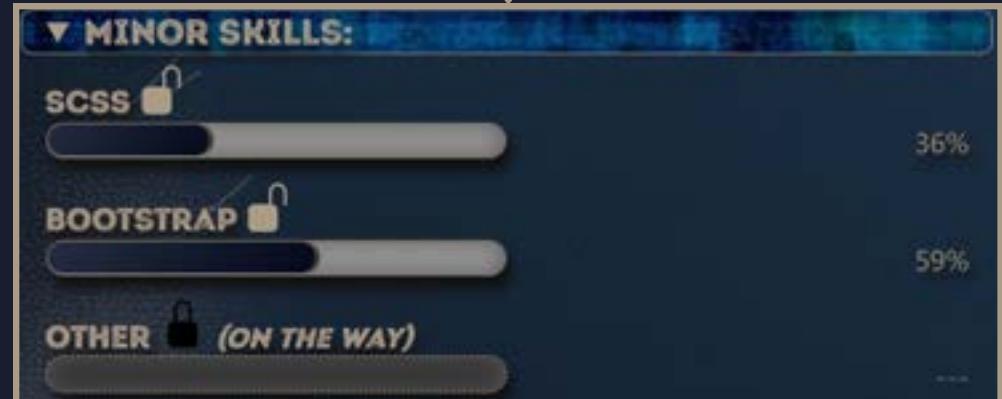
As simple as it might seem to be, this is one of the first most useful skills I found in my life, eventually facilitating my English language learning skill and better understanding more technical subjects such as math, project drawings and base IT in the later future.

**PROJECT PLANNING**

Context: High School  
Date: 2013

```
<details class="min-skills">
  <summary>Minor Skills:</summary>
  <ul
    class="min-skills-ord"
    role="list">
    <li class="min-skill-item">
      <h5>
        SCSS
        
      </h5>
      <div class="min-skill-cont">
        <div class="skill-bar">
          <span class="skill-mastery"></span>
          <span class="skill-bns"></span>
        </div>
        <p>36%</p>
      </div>
    </li>
  </ul>
</details>
```

Da un lato più tecnico, ho deciso di sfruttare i tag HTML `<details>` per rendere alcune sezioni facoltative da visualizzare, essendo più precise su eventuali abilità mirate (**Minor Skills**) o più descrittive (**Skills History**). Per il resto, i vari contenitori erano stati predisposti per `<ul>` e `<i>`.



# MAIN.MY-SKILLS

Come ultima cosa, a causa della pagina piuttosto corta (da chiusa), decisi inizialmente di eliminare il pulsante back-to-top, poi lo resi disattivato nei viewport più grandi e riattivato con altezza del contenitore minore rispetto alla pagina index.



# I\_AM\_ERROR (MAIN BONUS)

```
.i-am-error {  
background: black;  
display: flex;  
flex-direction: column;  
font-family: $font-pixel;  
padding: 5vh 2vw;  
height: 100vh;  
width: 100%;  
}  
  
h1 {  
color: darkslategray;  
font-family: $font-i-am-error;  
font-size: $h1-bonus;  
text-align: start;  
text-shadow: none;  
width: 50%;  
}  
  
q {  
color: $altered-text;  
font-size: $h2;  
font-style: italic;  
margin-top: 2vh;  
width: 50%;  
}  
  
em {  
font-size: $h3;  
font-style: normal;  
color: darkslategray;  
}  
em {  
color: $altered-text;  
font-weight: bold;  
}  
  
p {  
color: $title-color;  
margin-top: 2vh;  
text-indent: 0;  
text-shadow: none;  
width: 50%;  
}  
a {  
font-weight: bold;  
text-decoration: none;  
}:hover {  
color: darkslategray;  
}  
:focus {  
color: $altered-text;  
}
```

# I\_AM\_ERROR!

**“YOU’VE MET WITH A TERRIBLE FATE,  
HAVEN’T YOU?”**  
[QUOTE *HAPPY MASK SALESMAN*]

BUT DON’T YOU WORRY, YOU CAN RETURN BACK THROUGH THIS [LINK](#).  
ESPECIALLY BECAUSE THERE’S NOTHING HERE. YET. YOU’D GET BORED.  
THOUGH I STILL HOPE YOU CAUGHT THE QUOTE AND LIKED MY  
TEMPORARY JOKE.

Come ultimo tocco, ho occupato la sezione **My Projects** con una pagina temporanea contenente due citazioni nerd.  
A livello tecnico richiedeva poche righe di codice (fra cui `<q>`) e dei nuovi gruppi font registrati in `_base-stats`. Inoltre,  
una soddisfazione personale, riutilizzai un colore che per molto tempo tenevo predefinito durante la mia  
programmazione e che, successivamente, cadde in disuso: dark-slate gray.

```
<body>  
  <main class="i-am-error">  
    <h1>I_AM_ERROR!</h1>  
    <q>You've met with a terrible fate, haven't you?</q>  
    <em>[Quote <i>Happy Mask Salesman</i>]</em>  
    <p>But don't you worry, you can return back through this <a href="index.html">link</a>. Especially because there's nothing here. Yet. You'd get bored. Though I still hope you caught the quote and liked my temporary joke.</p>  
  </main>  
</body>
```

# HTML IN ITALIANO

SU DI ME

LE MIE ABILITÀ PERSONAGGIO

I MIEI PROGETTI

## JAVASCRIPT (IN CORSO) \*

O

\* Certe operazioni richiedevano l'uso di alcuni script per funzionare, come Bootstrap o EmailJS

## CONTATTAMI!

**NOME COMPLETO \***

Basch Von Ronseburg

**EMAIL \***

esempio@email.com

**IL TUO MESSAGGIO \***

Scrivi qui il tuo messaggio...

Invia

Resetta

Per ogni pagina HTML, ho preparato una copia esatta delle stesse e tradotto tutti tag contenenti descrizioni e paragrafi: `<img alt>`, `<p>`, `<h1-6>`, `<q>`, `<label>` e valori interni degli `<input>` del form e molto altro. Ho solo aggiunto un dettaglio...

# HTML IN ITALIANO

... per problemi di spaziatura nella sezione **Abilità Accessorie** nel viewport a 280px, ho aggiunto una classe *i.ita-br* per spezzare le parole, nascosta con *display: none* al di sopra di quel viewport.

```
<li class="access-skills-item">
  <h6>Pianifica<i class="ita-br">-</i>zione progettuale<i class="ita-br">-</i>le</h6>
  <p>
    Facilitazione della concentrazione progettuale dividendo il lavoro in punti di controllo multipli
    <em>+1 Slot Tattici</em>
  </p>
</li>
<li class="access-skills-item">
  <h6>Motivazione del team</h6>
  <p>
    Migliora il morale altrui
    <em>+1 ai Tiri Carisma</em>
  </p>
</li>
```

**PIANIFICA-ZIONE PROGETTUALE** Facilitazione della concentrazione progettuale dividendo il lavoro in punti di controllo multipli  
+ 1 Slot Tattici

**MOTIVAZIONE DEL TEAM** Migliora il morale altrui  
+ 1 ai Tiri  
Carisma

# META TAGS/OPEN GRAPH

```
<meta
  name="description"
  content="These are my skills learned thanks to Start2Impact course and previous works." />
<meta
  name="generator"
  content="Visual Studio Code" />
<meta
  name="keywords"
  content="html css scss personal webpage emailjs curriculum social links" />
<meta
  name="robots"
  content="all" />
<meta
  name="og:description"
  property="og:description"
  content="These are my skills learned thanks to Start2Impact course and previous works." />
<meta
  name="og:determiner"
  property="og:determiner"
  content="" />
<meta
  name="og:image"
  property="og:image"
  content="assets/img/omega-logo.png" />
<meta
  name="og:locale"
  property="og:locale"
  content="en_GB"
  href="my-skills.html" />
<meta
  name="og:locale:alternate"
  property="og:locale:alternate"
  content="it_IT"
  href="my-skills-it.html" />
<meta
  name="og:title"
  property="og:title"
  content="Omega: My skills" />
<meta
  name="og:site_name"
  property="og:site_name"
  content="Omega Webpage" />
<meta
  name="og:type"
  property="og:type"
  content="website" />
<meta
  name="og:url"
  property="og:url"
  content="index.html" />
<meta
  name="viewport"
  content="width=device-width, initial-scale=1.0, maximum-scale=5.0, user-scalable=1" />
```

Ultimi, ma non meno importanti, ho incluso tutti i possibili tag `<meta>` e *Open Graph* che potrebbero sintetizzare le mie pagine internet nei motori di ricerca. Non vi è molto da presentare se non direttamente inclusi nei codici stessi, che generalmente si ripetono.

*E nuovamente, si ringrazia la **Referente Sviluppo Patrizia** per alcune delucidazioni sul funzionamento della proprietà content*

# GITHUB PAGES

GitHub Pages



Completato lo sviluppo del progetto, ho seguito le indicazioni per caricare il mio sito internet online tramite **GitHub Pages**, nel quale dovetti cambiare il nome *project-html-css-emanuele-romano* in *omegaer.github.io*, includendo anche un file *README.md* nella repository di **GitHub** con il link diretto alla mia pagina.

# CONCLUSIONI

Posso notare un'enorme differenza fra il progetto introduttivo e quello di HTML & CSS, ma la mia strada è ancora all'inizio e non so ancora immaginare cosa mi riservino questi studi. So anche di essere ancora all'oscuro di tanti tag e combinazioni di stile che sicuramente mi avrebbero facilitato il lavoro su molte fasi progettuali, ma la mia intenzione è di curare nel dettaglio la mia pagina web anche al di fuori di un progetto di studio, trovando sicuramente sempre nuovi metodi per migliorare a livello estetico e strutturale e, magari, renderlo più appetibile ad aziende e motori di ricerca. Nel frattempo...



SALVE. PERMETTETEMI DI  
PRESENTARMI. IL MIO NOME È  
OMEGA.  
GRAZIE PER L'ATTENZIONE!