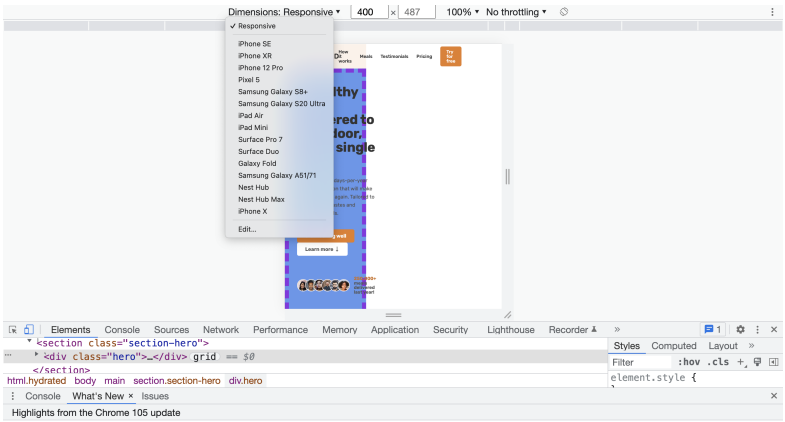
Media Queries(Assessibility)

- How Media Queries Work?
  - With max-width: **@media(max-width:600px)**, in Global CSS

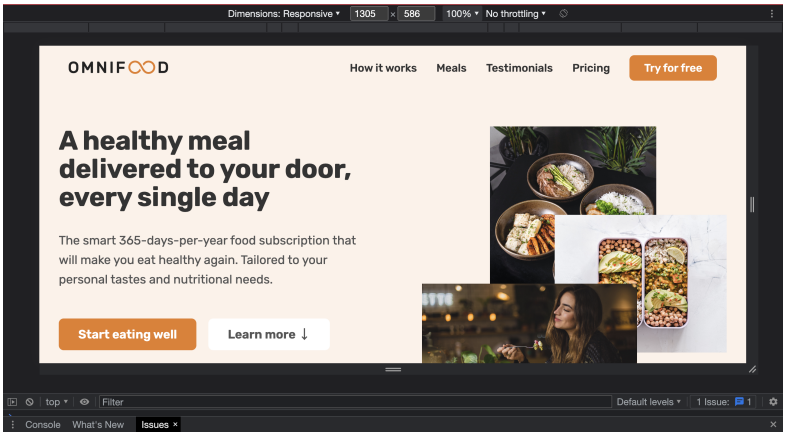
```
@media(max-width:1200px){
  .section-hero{
    background-color: orangered;
  }
}

@media(max-width:600px){
  .section-hero{
    border:20px dashed blueviolet;
    background-color: cornflowerblue;
  }
}
```

  - The code is efficient within 0~600px, when the width larger than 600px, the media query not apply
  - when there exists a conflict if the viewpoint multiple queries apply, finally the query with smallest range becomes effective. If there doesn't exist a conflict, then all of them make effect.
  - For different devices, we can select from *"Inspect"*



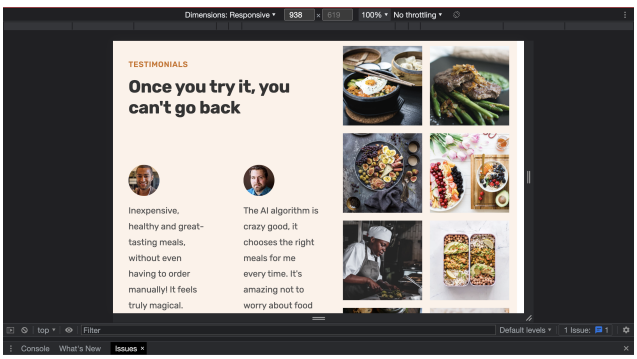
- How to select the Breakpoints?
  - Bad: based on popular devices: we set breakpoint between different device
  - Good Strategy: based on width ranges, set breakpoints between similar devices sizes 600, 900, 1200px
  - Perfect Strategy (quite hard completely use it): when design breaks down, only look at our content and design, ignore our device and width range
- Responding to Small Laptop(1200~1344px)
  - this line of code tells the viewpoint for responsive web design to match the device width do not need to zoom out(100%)

```
<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <!-- always include this line of code -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
```



- we only change the part we need in queries.css file, and we need to go back to original html and style/general.css files finding the corresponding codes.
  - **Frequently changed properties**
    - gap
    - padding
    - font-size
    - grid-templete-column/row
    - ……
- the calculation of **max-width(em)=breakpoint(px)/16**

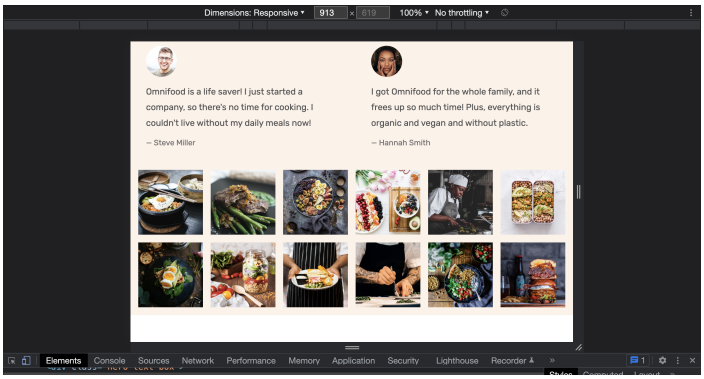
- Responding to Landscape Tablet(900~1200px)

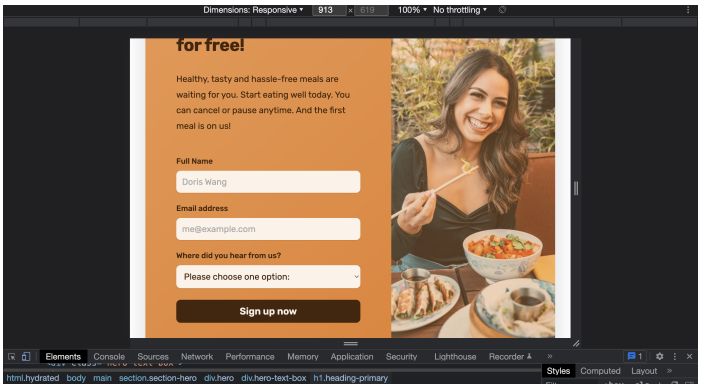


- if we want change the overall size by in proportion based on the original size(the paragraph default is 1.6rem, we want to set 9px font-size in this case), the percentage expression is needed in html element

```
@media (max-width:75em){
  html{
    /* 9/16px */
    font-size:56.25%;
  }
  .heading-secondary{
    font-size:3.6rem;
  }
  .heading-tertiary{
    font-size:2.4rem;
  }
}
```

- Responding to Tablet(704~944px)
  - In this case, many grid need to cut down its column's number using ***{grid-templete-columns: …;}***



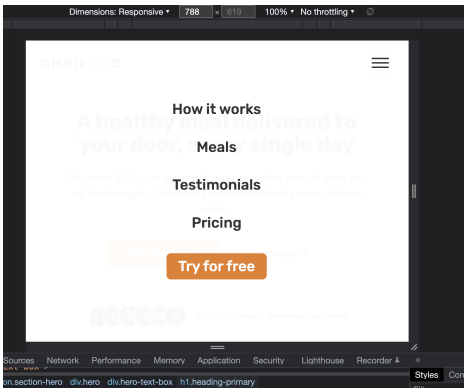


- Buliding the Mobile Navigation
  - if we just need to hide one button and do not want to add one class to sepicify,we can use this trik

```
<button class="btn-mobile-nav">
  <ion-icon class="icon-mobile-nav" name="menu-outline"></ion-icon>
  <ion-icon class="icon-mobile-nav" name="close-outline"></ion-icon>
</button>
```

```
.icon-mobile-nav[name="close-outline"]{
  display:none;
}
```

  - Hide and Show Issues
    - Hide the mobile navigation without js file by manual, we set ***{opacity:0;}***
    - if we open the navigation manually, we add "nav-open" class in header, and set it and nav-main to ***{opacity:1;}***



    - 3 steps only shows on mobile

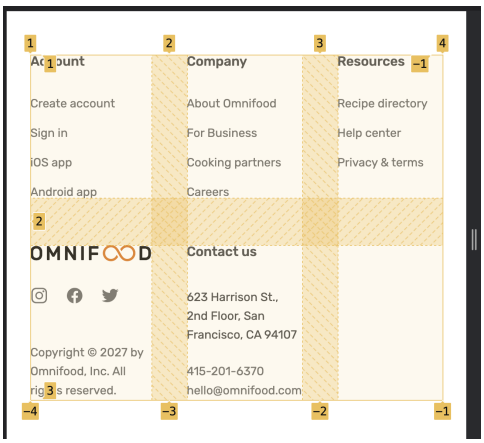
```
  /* 1)hide it visually */
  opacity:0;
  /* 2)make it unaccessable to mouse and keybroad */
  pointer-events:none;
  /* 3)hider it from screen readers */
  visibility: hidden;
}

.nav-open .main-nav{
  opacity:1;
  pointer-events: auto;
  visibility:visible;
}
```

- we move the whole nav to the right of hero section use ***{transform:translateX(100%);}***

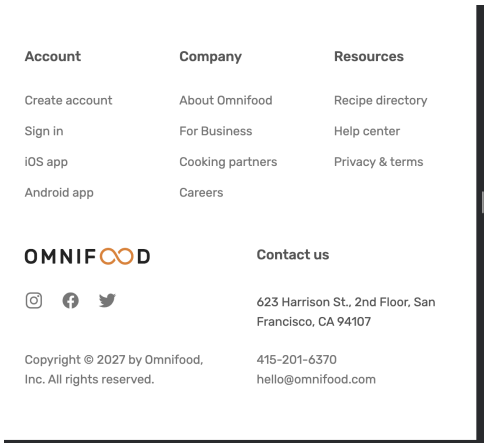
- Responding to Smaller Tablet(500~704px)
  - we change the 3 grid column into (2, 1fr) format, we want the last one to span over the column and center the text

```
@media(max-width:44em){
  .grid--3-cols,
  .grid--4-cols {
    grid-template-columns: repeat(2, 1fr);
  }
  .diets{
    /* span the column all the way to the end of the grid */
    grid-column: 1/-1;
    justify-self: center;
  }
```

  - we want to change the last row with 2 columns items to span over the whole row,we need to change the grid into 6 columns. For the first row each item spans for 2 columns, for the last row each item spans for 3columns

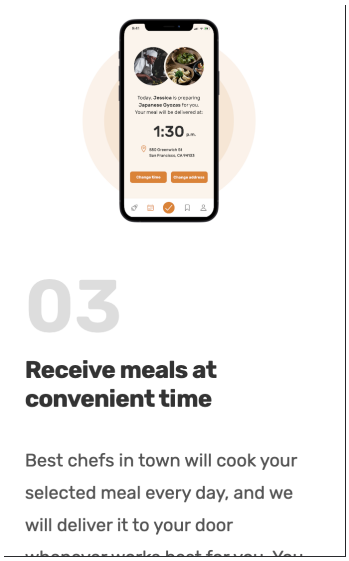



```
.grid--footer {
  grid-template-columns: repeat(6,1fr);
}
.logo-col,
.address-col{
  grid-column: span 3;
}
.nav-col{
  grid-row:1;
  grid-column: span 2;
}
```



- Responding to Phone(300~704px)
  - When comes to phone screen, we should set all grid column number to **1**.
  - reorder the elements in F-style, we put all photos above the texts, use ***grid-row***

```css
.step-img-box:nth-child(2){
  grid-row: 1;
}
.step-img-box:nth-child(6){
  grid-row: 5;
}
```



以上内容整理于 幕布文档