1. Components
2. HomeComponent

## File

src/app/components/home/home.component.ts

## Description

Component for home page

## Implements

_OnInit_

## Metadata

| selector | app-home |
|---|---|
| styleUrls | ./home.component.scss |
| templateUrl | ./home.component.html |

## Index

**Properties**

- Public bestD
- Public bestM
- Public bestOther No
- Public bestT
- Public bestTeamHeroes
- Public idParams
- images
- Public mostF
- Public moststarts
- Public mostStatsHero
- Public newHeros
- Public stats

**Methods**

- bestDc
- bestMarvel
- bestOTHero
- bestTeam
- herosMost
- mostFollwed
- newHeroesAdd
- ngOnInit

## Constructor

```
constructor(_hero: HeroService, _UserHero: UserHeroService, _Team: TeamService)
```
Defined in [src/app/components/home/home.component.ts:71](#)

Constructor in which we inject hero service, formBuilder and rouser service

**Parameters :**

| Name | Type | Optional |
|---|---|---|
| _hero | HeroService | No |
| _UserHero | UserHeroService | No |
| _Team | TeamService | No |

## Methods

**bestDc**
```
bestDc()
```
Defined in [src/app/components/home/home.component.ts:132](#)

Bring the best Dc hero

**Returns :** _void_

**bestMarvel**
```
bestMarvel()
```
Defined in [src/app/components/home/home.component.ts:101](#)

Bring the best marvel hero

**Returns :** _void_

**bestOTHero**
```
bestOTHero()
```
Defined in [src/app/components/home/home.component.ts:117](#)

Bring the best other hero

**Returns :** _void_

**bestTeam**
```
bestTeam()
```
Defined in [src/app/components/home/home.component.ts:161](#)

Bring the team with highest stats

**Returns :** <u>void</u>

**herosMost**

```
herosMost()
```

Defined in <u>src/app/components/home/home.component.ts:176</u>

Bring the heroes with the highest stats

**Returns :** <u>void</u>

**mostFollwed**

```
mostFollwed()
```

Defined in <u>src/app/components/home/home.component.ts:147</u>

Bring the most followed hero

**Returns :** <u>void</u>

**newHeroesAdd**

```
newHeroesAdd()
```

Defined in <u>src/app/components/home/home.component.ts:192</u>

Bring the newest heroes in the db

**Returns :** <u>void</u>

**ngOnInit**

```
ngOnInit()
```

Defined in <u>src/app/components/home/home.component.ts:87</u>

Start when the component inits

**Returns :** <u>void</u>

## Properties

**Public bestD**

*Type :* <u>any</u>
*Default value :* `''`
Defined in <u>src/app/components/home/home.component.ts:25</u>

variable to store best DC hero

**Public bestM**

*Type :* <u>any</u>
*Default value :* `''`
Defined in <u>src/app/components/home/home.component.ts:21</u>

variable to store best Marvel hero

**Public bestOther**

*Type :* <u>any</u>
*Default value :* `''`
Defined in <u>src/app/components/home/home.component.ts:29</u>

variable to store best other hero

**Public bestT**

*Type :* <u>any</u>
Defined in <u>src/app/components/home/home.component.ts:37</u>

variable to store best team

**Public bestTeamHeroes**

*Type :* <u>Hero[]</u>
*Default value :* `[]`
Defined in <u>src/app/components/home/home.component.ts:41</u>

variable to store best team heroes

**Public idParams**

*Type :* <u>number</u>
*Default value :* `0`
Defined in <u>src/app/components/home/home.component.ts:71</u>

variable to save the id from params

**images**

*Default value :* `['photo_10.jpg', 'photo_9.jpg', 'photo_7.jpg'].map((n) => `../../../assets/img/${n}`)`
Defined in <u>src/app/components/home/home.component.ts:97</u>

**Public mostF**

*Type :* <u>Hero[]</u>
Defined in <u>src/app/components/home/home.component.ts:33</u>

variable to store most followed heroes

**Public moststarts**

*Type :* <u>Hero[]</u>
*Default value :* `[]`
Defined in <u>src/app/components/home/home.component.ts:54</u>

variable to store heroes with more votes

**Public mostStatsHero**

*Type :* <u>Hero[]</u>
*Default value :* `[]`
Defined in <u>src/app/components/home/home.component.ts:45</u>

variable to store heroes with best stats

**Public newHeros**

*Type :* [Hero[]](#)

*Default value :* `[]`

Defined in [src/app/components/home/home.component.ts:49](#)

variable to store new heros in bd

**Public stats**

*Type :* `string[]`

*Default value :* `[ 'mostIntelligence', 'mostStrength', 'mostSpeed', 'mostDurability', 'mostPower', 'mostCombat' ]`

Defined in [src/app/components/home/home.component.ts:59](#)

variable to store hero stats

```
import { Component, OnInit } from '@angular/core'
import { HeroService } from 'src/app/services/hero.service'
import { UserHeroService } from 'src/app/services/user-hero.service'
import { TeamService } from 'src/app/services/team.service'
import { Team } from 'src/app/models/team'
import { Hero } from 'src/app/models/hero'

/**
 * Component for home page
 */
@Component({
  selector: 'app-home',
  templateUrl: './home.component.html',
  styleUrls: ['./home.component.scss']
})

export class HomeComponent implements OnInit {
  /**
   * variable to store best Marvel hero
   */
  public bestM: any = ''
  /**
   * variable to store best DC hero
   */
  public bestD: any = ''
  /**
   * variable to store best other hero
   */
  public bestOther: any = ''
  /**
   * variable to store most followed heroes
   */
  public mostF: Hero[]
  /**
   * variable to store best team
   */
  public bestT: any;
  /** variable to store best team heroes
   *
   */
  public bestTeamHeroes: Hero[] = []
  /**
   * variable to store heroes with best stats
   */
  public mostStatsHero: Hero[] = []
  /**
   * variable to store new heros in bd
   */
  public newHeros: Hero[] = []

  /**
   * variable to store heroes with more votes
   */
  public moststarts: Hero[] = []

  /**
   * variable to store hero stats
   */
  public stats: string[] = [
    'mostIntelligence',
    'mostStrength',
    'mostSpeed',
    'mostDurability',
    'mostPower',
    'mostCombat'
  ]

  /**
   * variable to save the id from params
   */
  public idParams: number = 0

  /**
   * Constructor in which we inject hero service, formBuilder and rouser service
   */
  constructor(
    private _hero: HeroService,
    private _UserHero: UserHeroService,
    private _Team: TeamService
  ) {
    this.bestT = new Object
  }

  /**
   * Start when the component inits
   */
  ngOnInit() {
    this.bestMarvel()
    this.bestDc()
    this.mostFollwed()
    this.bestTeam()
    this.bestOTHero()
    this.herosMost()
    this.newHeroesAdd()
  }

  images = ['photo_10.jpg', 'photo_9.jpg', 'photo_7.jpg'].map((n) => `../../../assets/img/${n}`);
  /**
   * Bring the best marvel hero
   */
  bestMarvel(): void {
    window.scrollTo(0, 0);
    this._UserHero.bestMarverHero().subscribe(
      res => {
        this.bestM = res[0]
        this.moststarts.push(this.bestM)
      },
      error => {
        console.log(error)
      }
    )
  }

  /**
   * Bring the best other hero
   */
  bestOTHero(): void {
    this._UserHero.bestOTHero().subscribe(
      res => {
```

```
          this.bestOther = res[0]
          this.moststarts.push(this.bestOther)
        },
        error => {
          console.log(error)
        }
      )
  }

  /**
   * Bring the best Dc hero
   */
  bestDc(): void {
    this._UserHero.bestDCHero().subscribe(
      res => {
        this.bestD = res[0]
        this.moststarts.push(this.bestD)
      },
      error => {
        console.log(error)
      }
    )
  }

  /**
   * Bring the most followed hero
   */
  mostFollwed(): void {
    this._UserHero.mostFollowHeros().subscribe(
      res => {
        this.mostF = res
      },
      error => {
        console.log(error)
      }
    )
  }

  /**
   * Bring the team with highest stats
   */
  bestTeam(): void {
    this._Team.bestTeam().subscribe(
      res => {
        this.bestT = res[0]
        this.idParams = this.bestT.idUsu
      },
      error => {
        console.log(error)
      }
    )
  }

  /**
   * Bring the heroes with the highest stats
   */
  herosMost(): void {
    this.stats.forEach(stats => {
      this._hero[stats]().subscribe(
        res => {
          this.mostStatsHero.push(res)
        },
        error => {
          console.log(error)
        }
      )
    })
  }

  /**
   * Bring the newest heroes in the db
   */
  newHeroesAdd(): void {
    this._hero.newHeros().subscribe(
      res => {
        this.newHeros = res
      },
      error => {
        console.log(error)
      }
    )
  }
}

<div id="container">
  <div class="carrousel">
    <ngb-carousel *ngIf="images">
      <ng-template ngbSlide>
        <div class="picsum-img-wrapper">
          <img [src]="images[0]" alt="Random first slide" />
        </div>
        <div class="carousel-caption">
          <h3>Hero recomended</h3>
          <p>Search for all kind of heroes and learn new things about them</p>
        </div>
      </ng-template>
      <ng-template ngbSlide>
        <div class="picsum-img-wrapper">
          <img [src]="images[1]" alt="Random second slide" />
        </div>
        <div class="carousel-caption">
          <h3>EPIC BATTLE!!! ONE VS ONE</h3>
          <p>Fight one by one to see who is the best hero!</p>
        </div>
      </ng-template>
      <ng-template ngbSlide>
        <div class="picsum-img-wrapper">
          <img [src]="images[2]" alt="Random third slide" />
        </div>
        <div class="carousel-caption">
          <h3>BATTLE OF TEAMS</h3>
          <p>Create you OWN team and fight with other users</p>
        </div>
      </ng-template>
    </ngb-carousel>
  </div>
  <div class="color1">
    <h3 class="title">BEST HEROES BY PUBLISHER</h3>
    <div id="MarvelVsDC">
      <div class="card1" *ngFor="let bestM of moststarts">
        <div class="imgCircle">
          <img
            [src]="bestM.image"
            onError="this.src='../../../assets/img/nogender_2.jpg'"
            [routerLink]="['/heroesDetail', bestM.idHero]"
            [alt]="bestM.heroName"
            class="img-fluid"
          />
        </div>
        <div class="points">
          <p *ngIf="bestM.score < 10">{{ bestM.score }}</p>
          <p *ngIf="bestM.score == 10" class="ten">{{ bestM.score }}</p>
          <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 26 26">
            <g transform="translate(0 -1026.362)">
              <circle cx="13" cy="1039.362" r="13" fill="#ffa40b" />
              <path
                style="
                  line-height: normal;
                  text-indent: 0;
```

```
                  text-align: start;
                  text-decoration-line: none;
                  text-decoration-style: solid;
                  text-decoration-color: #000;
                  text-transform: none;
                  block-progression: tb;
                  isolation: auto;
                  mix-blend-mode: normal;
                  marker: none;
                "
                fill="#ff7712"
                d="M 15.042969 25.830078 A 12.999989 12.999989 0 0 0 15.208984 25.810547 A 12.999989 12.999989 0 0 0 16.476562 25.525391 A 12.999989 12.999989 0 0 0 17.710938 25.11718
                color="#000"
                font-family="sans-serif"
                font-weight="400"
                transform="translate(0 1026.362)"
                white-space="normal"
              />
              <path
                style="
                  line-height: normal;
                  text-indent: 0;
                  text-align: start;
                  text-decoration-line: none;
                  text-decoration-style: solid;
                  text-decoration-color: #000;
                  text-transform: none;
                  block-progression: tb;
                  isolation: auto;
                  mix-blend-mode: normal;
                  marker: none;
                "
                fill="#fff"
                d="m 78.795036,1015.7352 c 0.05089,0.074 0.09807,0.137 0.162364,0.2547 0.194225,0.3554 0.422345,0.868 0.647368,1.3973 0.225027,0.5294 0.448538,1.0769 0.652305,1.529 0.
                color="#000"
                font-family="sans-serif"
                font-weight="400"
                overflow="visible"
                transform="matrix(1.06795 0 0 .97117 -71.11 46.368)"
                white-space="normal"
              />
            </g>
          </svg>
        </div>
        <div class="infoHero">
          <h2>{{ bestM.heroName }}</h2>
          <h4>{{ bestM.publisher }}</h4>
        </div>
        <div class="semiCircle">
          <div class="dots">
            <div class="dot1"></div>
            <div class="dot2"></div>
            <div class="dot3"></div>
            <div class="dot4"></div>
            <div class="dot4"></div>
            <div class="dot5"></div>
            <div class="dot6"></div>
            <div class="dot7"></div>
            <div class="dot8"></div>
          </div>
          <div class="dots2">
            <div class="dot1"></div>
            <div class="dot2"></div>
            <div class="dot3"></div>
            <div class="dot4"></div>
            <div class="dot4"></div>
            <div class="dot5"></div>
            <div class="dot6"></div>
            <div class="dot7"></div>
            <div class="dot8"></div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

<div class="color2">
  <h3 class="title">MOST FOLLOWED</h3>
  <div id="mostF">
    <div class="card2" *ngFor="let hero of mostF">
      <img
        [src]="hero.image"
        onError="this.src='../../../assets/img/nogender_2.jpg'"
        [routerLink]="['/heroesDetail', hero.idHero]"
        [alt]="hero.heroName"
        class="img-fluid img2"
      />
      <div class="overImg">
        <h4>{{ hero.heroName }}</h4>
      </div>
      <div class="iconoUser">
        <p>{{ hero.followers }}</p>
        <svg
          xmlns="http://www.w3.org/2000/svg"
          width="48"
          height="48"
          viewBox="0 0 48 48"
        >
          <g transform="translate(0 -1004.362)">
            <circle cx="24" cy="1028.362" r="24" fill="#1d67b1" />
            <path
              fill="#154082"
              fill-rule="evenodd"
              style="
                line-height: normal;
                text-indent: 0;
                text-align: start;
                text-decoration-line: none;
                text-decoration-style: solid;
                text-decoration-color: #000;
                text-transform: none;
                block-progression: tb;
                isolation: auto;
                mix-blend-mode: normal;
              "
              d="M 25.931641 47.910156 A 23.999992 23.999992 0 0 0 28.080078 47.650391 A 23.999992 23.999992 0 0 0 30.419922 47.125 A 23.999992 23.999992 0 0 0 32.697266 46.369141 A
              color="#000"
              font-family="sans-serif"
              font-weight="400"
              transform="translate(0 1004.362)"
              white-space="normal"
            />
            <path
              fill="#fff"
              fill-rule="evenodd"
              style="
                line-height: normal;
                text-indent: 0;
                text-align: start;
                text-decoration-line: none;
                text-decoration-style: solid;
                text-decoration-color: #000;
                text-transform: none;
                block-progression: tb;
                isolation: auto;
                mix-blend-mode: normal;
              "
              d="m 15.480469,4.4335938 c -3.021901,0 -5.4805005,2.4692 -5.480469,5.5 2.9e-5,3.0308002 2.458611,5.5000002 5.480469,5.5000002 3.021858,0 5.48044,-2.4692 5.480469,-5.50
              color="#000"
```

```
                    font-family="sans-serif"
                    font-weight="400"
                    overflow="visible"
                    transform="translate(8.5 1011.362)"
                    white-space="normal"
                />
              </g>
            </svg>
          </div>
        </div>
      </div>
    </div>
  <div class="color1">
    <h3 class="title">
      BEST TEAM: <span class="Tred">{{ bestT.teamName }}</span>
    </h3>
    <div id="bestT">
      <app-team [idUsu]="idParams"></app-team>
      <h4>
        By:
        <a [routerLink]="['/publicProfile', bestT.idUsu]"
          ><span class="Tred">{{ bestT.alias }} </span></a
        >
      </h4>
      <h4>Total Points: {{ bestT.totalPoint }}</h4>
    </div>
  </div>
  <div class="color2">
    <h2 class="title">THE MOST POWERFUL</h2>
    <div id="TheBest">
      <div class="card2" *ngFor="let hero of mostStatsHero">
        <img
          [src]="hero.image"
          onError="this.src='../../../assets/img/nogender_2.jpg'"
          [routerLink]="['/heroesDetail', hero.idHero]"
          [alt]="hero.heroName"
          class="img-fluid img2"
        />
        <div class="overImg">
          <h4>{{ hero.heroName }}</h4>
        </div>
        <div class="iconM" *ngIf="hero.intelligence">
          <p>{{ hero.intelligence }}</p>
          <img src="../../../assets/img/intelligence.svg" alt="" />
        </div>
        <div class="iconM" *ngIf="hero.strength">
          <p>{{ hero.strength }}</p>
          <img src="../../../assets/img/strenght.svg" alt="" />
        </div>
        <div class="iconM" *ngIf="hero.speed">
          <p>{{ hero.speed }}</p>
          <img src="../../../assets/img/speed.svg" alt="" />
        </div>
        <div class="iconM" *ngIf="hero.durability">
          <p>{{ hero.durability }}</p>
          <img src="../../../assets/img/durability.svg" alt="" />
        </div>
        <div class="iconM" *ngIf="hero.power">
          <p>{{ hero.power }}</p>
          <img src="../../../assets/img/power.svg" alt="" />
        </div>

        <div class="iconM" *ngIf="hero.combat">
          <P> {{ hero.combat }}</P>
          <img src="../../../assets/img/combat.svg" alt="" />
        </div>
      </div>
    </div>
  </div>
  <div class="color1">
    <h3 class="title">LAST ADDED HEROES</h3>
    <div id="mostF">
      <div class="card2" *ngFor="let hero of newHeros">
        <img
          *ngIf="hero.image"
          [src]="hero.image"
          [routerLink]="['/heroesDetail', hero.idHero]"
          [alt]="hero.heroName"
          onError="this.src='../../../assets/img/nogender_2.jpg'"
          class="img-fluid img2"
        />
        <div class="overImg">
          <h4>{{ hero.heroName }}</h4>
        </div>
      </div>
    </div>
  </div>
</div>


./home.component.scss

#MarvelVsDC,
#mostF,
#TheBest {
  display: flex;
  flex-direction: row;
  justify-content: space-around;
}

#mostF,
#TheBest {
  flex-wrap: wrap;
}

.card1 {
  padding: 0.2em;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  text-align: center;
  box-shadow: 0 0 21px 9px rgba(8, 17, 10, 0.63);
  width: 7em;
  height: 10em;
  background-color: #272727;
  overflow: hidden;
  position: relative;
  border-radius: 2%;
}

.card1 .imgCircle {
  cursor: pointer;
  margin-bottom: 2em;
  border-radius: 50%;
  margin-top: 0.5em;
  overflow: hidden;
}

.semiCircle {
  width: 122%;
  height: 73%;
  bottom: -40%;
  z-index: 0;
  position: absolute;
  background: #00a23d;
  border-radius: 201px 201px 0 0;
  display: flex;
  flex-direction: row;
  justify-content: space-around;
```

```css
}

.dots {
  display: flex;
  flex-wrap: wrap;
  margin-right: 7em;
  width: 3.5em;
  height: 4em;
  padding-top: 5em;
}

.dots2 {
  display: flex;
  flex-wrap: wrap;
  width: 3.5em;
  height: 4em;
  padding-top: 5em;
}

.dots div,
.dots2 div {
  border-radius: 50%;
  width: 0.3em;
  margin: 0.4em;
  height: 0.3em;
  background-color: #272727;
}
.infoHero {
  font-family: "B612";
  height: 40%;
  width: 81%;
  z-index: 2;
  bottom: -17%;
  color: rgb(223, 221, 221);
  position: absolute;
}
.infoHero h2,
h4 {
  font-size: 0.5rem;
}

.card1:hover img {
  transform: scale(1.1);
  -ms-transform: scale(1.1);
  -moz-transform: scale(1.1);
  -webkit-transform: scale(1.1);
  -o-transform: scale(1.1);
}
.card1:hover .semiCircle {
  background: #cc4224;
}

.points {
  right: 0em;
  top: 0em;
  position: absolute;
}
.points svg {
  height: 2.5em;
}

.points p {
  position: absolute;
  font-family: "B612";
  font-size: 12px;
  right: 1.3em;
  top: 1em;
}
.points .ten {
  right: 0.9em;
}
.picsum-img-wrapper img {
  width: 100%;
}

.title {
  font-size: 1rem;
  color: #00a23d;
  font-family: "B612";
  padding-bottom: 1em;
}

.card2 {
  position: relative;
  width: 42%;
  padding-bottom: 1.5em;
}

.card2 .img2 {
  cursor: pointer;
  border-radius: 2%;
  box-shadow: 0 0 21px 9px rgba(8, 17, 10, 0.63);
  height: 60vw;
  width: 100%;
}
.overImg {
  height: 25%;
  background-color: #00a23eb6;
  width: 100%;
  bottom: 10%;
  padding-top: 7%;
  position: absolute;
  text-align: center;
  color: rgb(223, 221, 221);
}

.card2:hover .overImg {
  background: #cc4224;
}
.overImg h4 {
  font-size: 1rem;
}

.card2:hover .img2 {
  transform: scale(1.1);
  -ms-transform: scale(1.1);
  -moz-transform: scale(1.1);
  -webkit-transform: scale(1.1);
  -o-transform: scale(1.1);
}

#bestT {
  color: rgb(223, 221, 221);
}
#bestT h4 {
  text-align: right;
}

.iconoUser {
  z-index: 3;
  top: 2%;
  right: 2%;
  position: absolute;
}
.iconoUser p,
.iconM p {
  color: #cc4224;
  font-size: 4vw;
  right: 9vw;
```

```css
    font-family: "B612";
    position: absolute;
    text-shadow: -1px -1px 0 rgb(223, 221, 221), 1px -1px 0 rgb(223, 221, 221),
      -1px 1px 0 rgb(223, 221, 221), 1px 1px 0 rgb(223, 221, 221);
}
.Tred {
    color: #cc4224;
    font-family: "B612";
}
a {
    text-decoration: none;
}

.iconM {
    z-index: 3;
    top: 2%;
    right: 2%;
    position: absolute;
}
.iconM img {
    width: 12vw;
    height: 12vw;
}
#bestT h4 {
    font-size: 1rem;
}
.carousel-caption {
    display: none;
}

::ng-deep .carousel-indicators {
    display: none !important;
}
@media (min-width: 992px) {
    .title {
        font-size: 1.75rem;
    }
    ::ng-deep .carousel-indicators {
        display: flex !important;
    }

    .carousel-caption {
        display: inline;
        background-color: rgba(3, 0, 0, 0.288);
    }
    .carousel-caption h3,
    .carousel-caption p {
        color: #00a23d;
        font-family: "Bangers", cursive;
        text-shadow: -1px -1px 0 #fff, 1px -1px 0 #fff, -1px 1px 0 #fff,
          1px 1px 0 #fff;
        letter-spacing: 3px;
    }
    .iconM img {
        width: 4vw;
        height: 4vw;
    }
    .points {
        right: 1em;
        top: 2em;
    }

    .points svg {
        height: 4.2em;
    }
    .points p {
        font-size: 16px;
        right: 1.8em;
        top: 1.6em;
    }
    .points .ten {
        right: 1.5em;
    }
    .card1 {
        padding: 1.5em;
        font-family: "B612";
        width: 22em;
        height: 26em;
    }

    .card1 .imgCircle {
        margin-bottom: 5em;
        margin-top: 0em;
    }
    .card2 {
        width: 15%;
        font-family: "B612";
    }

    .card2 .img2 {
        height: 16vw;
    }
    .card2 h4 {
        font-size: 1.4vw;
    }

    #MarvelVsDC,
    #mostF,
    #TheBest {
        flex-direction: row;
    }

    .iconoUser p,
    .iconM p {
        right: 2vw;
        font-size: 1vw;
    }

    .infoHero h2 {
        font-size: 2rem;
    }

    .infoHero h4 {
        font-size: 1.5rem;
    }

    #bestT h4 {
        font-size: 1.5rem;
    }

    .overImg h4 {
        font-size: 1.5vw;
    }
}

@media (min-width: 768px) and (max-width: 991px) {
    .title {
        font-size: 1.75rem;
    }
    #bestT h4 {
        font-size: 1.5rem;
    }

    .points {
        right: 1em;
        top: 2em;
    }

    .points svg {
```

```
    height: 4.2em;
  }
  .points p {
    font-size: 16px;
    right: 1.8em;
    top: 1.6em;
  }
  .points .ten {
    right: 1.5em;
  }
  .card1 {
    padding: 1.5em;
    font-family: "B612";
    width: 15em;
    height: 19em;
  }

  .card1 .imgCircle {
    margin-bottom: 5em;
    margin-top: 0em;
  }

  .semiCircle {
    width: 105%;
    height: 73%;
    bottom: -29%;
  }

  .infoHero h2,
  h4 {
    font-size: 1.5rem;
  }

  .overImg h4 {
    font-size: 4vw;
  }
}
```

**Legend**
Html element
Component
Html element with directive

# result-matching ""

# No results matching ""