

Material-UI

React components that implement
Google's Material Design

Created By: David Yakin 2022

תוכן עניינים

4	<u>Introduction</u>	▪
5	<u>Installation</u>	▪
6	<u>Adding Roboto Fonts</u>	▪
10	<u>Components</u>	▪
11	<u>Data Display</u>	▪
12	<u>Typography</u>	▪
21	<u>sx prop</u>	▪
23	<u>Divider</u>	▪
27	<u>Button</u>	▪
33	<u>Card</u>	▪
42	<u>List</u>	▪
51	<u>Layout</u>	▪
52	<u>Box</u>	▪
55	<u>Container</u>	▪
60	<u>Stack</u>	▪
57	<u>Grid</u>	▪
74	<u>Navigation</u>	▪
75	<u>AppBar</u>	▪
78	<u>Bottom Navigation</u>	▪

100

Form ■



Benefits

Browsers compatibility

Responsive grid system

Wide customization options

Focus on aesthetically pleasing website

React based components

Comprehensible documentation

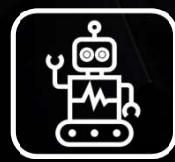




Installation

```
npm install @mui/material @emotion/react  
@emotion/styled @mui/icons-material  
@fontsource/roboto
```





Adding Roboto Fonts

על מנת שהעיצוב של האתר שלנו יהיה אחיד
הורדנו את הフォント Roboto שוממלץ לעבודה
עם הספרייה UI material-UI



Index.js

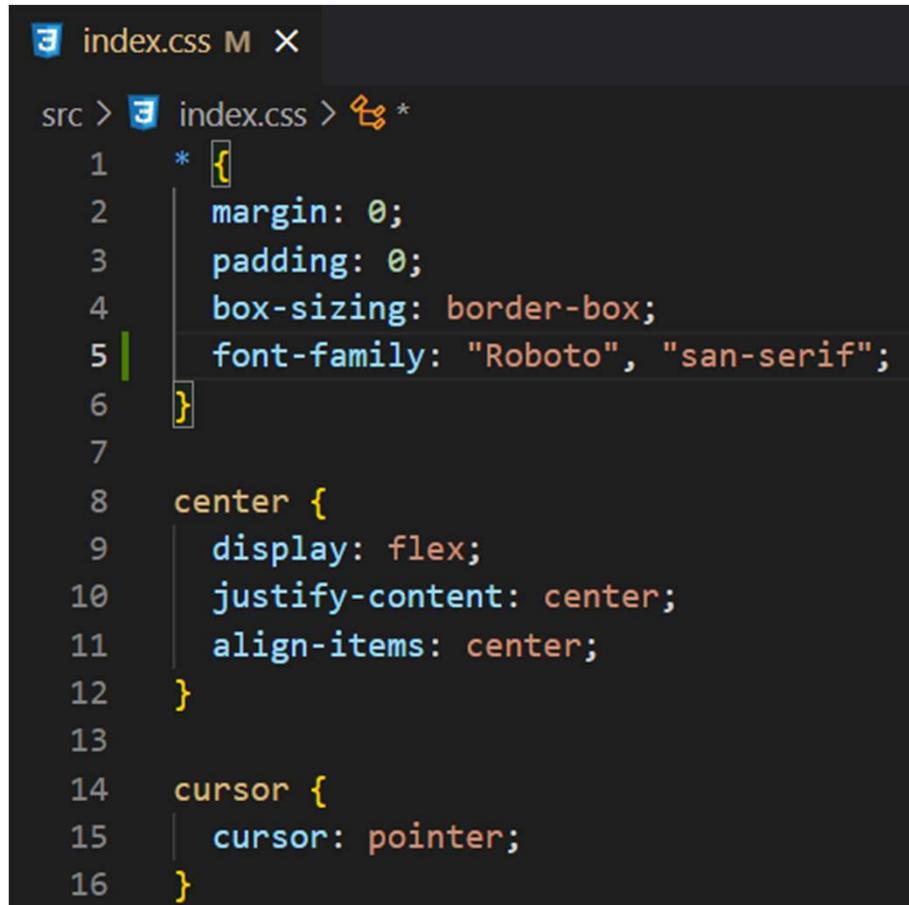
לאחר שהורדנו את הפונט של Roboto علينا לעדכן אותו בפונט הדיפולטיבי של האפליקציה

- ניבא את כל גדי הפונט לקובץ הבסיסה לאפליקציה (Entry point)

```
JS index.js M ×  
src > JS index.js > ...  
1 import React from "react";  
2 import ReactDOM from "react-dom/client";  
3 import "./index.css";  
4 import App from "./App";  
5 import reportWebVitals from "./reportWebVitals";  
6  
7 import "@fontsource/roboto/300.css";  
8 import "@fontsource/roboto/400.css";  
9 import "@fontsource/roboto/500.css";  
10 import "@fontsource/roboto/700.css";  
11  
12 const root = ReactDOM.createRoot(document.getElementById("root"));  
13 root.render(  
14   <React.StrictMode>  
15     | <App />  
16   </React.StrictMode>  
17 );  
18  
19 // If you want to start measuring performance in your app, pass a  
function  
20 // to log results (for example: reportWebVitals(console.log))  
21 // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals  
22 reportWebVitals();
```

Index.css

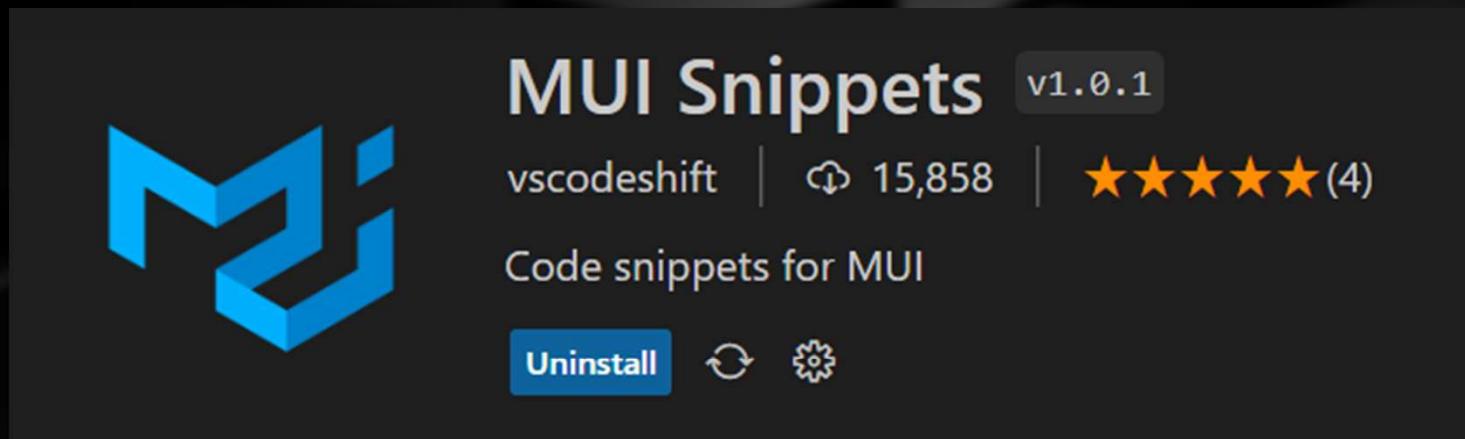
- נוסיף את הפונט של Roboto לעיצוב הגלובלי של האפליקציה



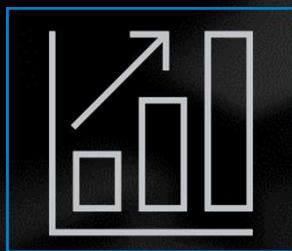
```
index.css M X
src > index.css > * *
1  * {
2    margin: 0;
3    padding: 0;
4    box-sizing: border-box;
5    font-family: "Roboto", "sans-serif";
6  }
7
8  center {
9    display: flex;
10   justify-content: center;
11   align-items: center;
12 }
13
14  cursor {
15   cursor: pointer;
16 }
```

MUI Snippets

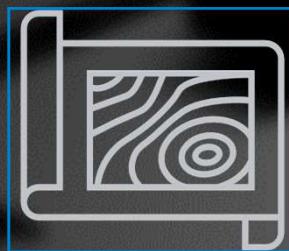
תוסף שיעזר לנו בעבודה עם MUI בסביבת העבודה של vscode



Components



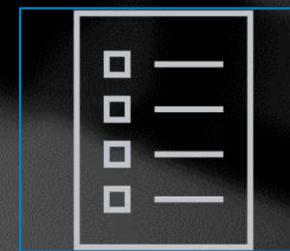
Data display &
feedback



Layout



Navigation



Form & Inputs



Data Display

קומפוננטות שעוסקות בתצוגת מידע בדרכים שונות לגולש





Typography

Too many type sizes and styles at once can spoil any layout. A typographic scale has a limited set of type sizes that work well together along with the layout grid.

<https://mui.com/material-ui/customization/typography>



Typography

קומפוננט ש אחראי על עיצוב של טקסט

- ניבא את הקומפוננט לתוכן קבוע בשם Typography מתוך ספריית UI M שהורדנו לפרויקט

- נציב את הקומפוננט במקום המועד ל – HTML

- התוצאה בדף:

▪ ניתן לראות שהטקסט נכתב ביחד עם הערכים הדיפולטיביים של MUI

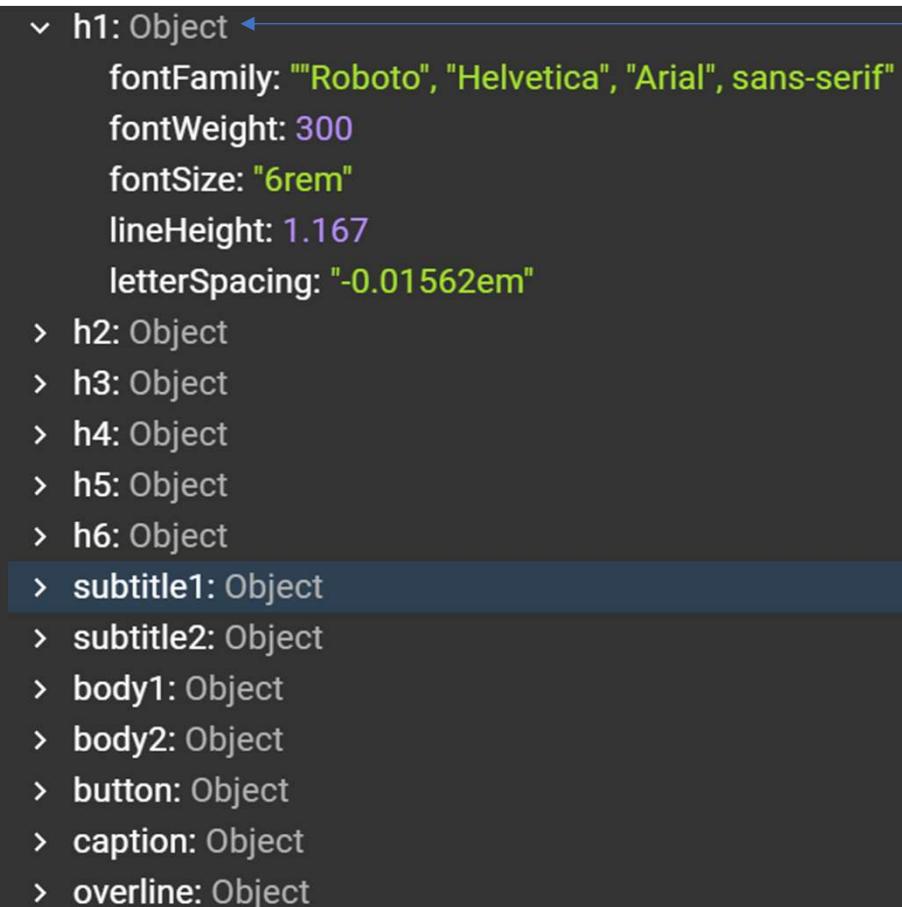
TypographyComponent.jsx

```
material-ui-sandbox > src > components > TypographyComponent.jsx > ...
1 import React from "react";
2 import Typography from "@mui/material/Typography"; ←
3
4 const TypographyComponent = () => {
5   return <Typography>One</Typography>; ←
6 }
7
8 export default TypographyComponent;
```

One

! יש שני דרכים ליבא קומפוננט זאת האחת היא כפי שמוצג לעיל והשנייה היא על ידי חילוץ הקומפוננט מתוך
@mui/material

```
8 const TypographyComponent = () => {
9   return <Typography variant="h4">Two</Typography>;
10};
```



```
h1: Object
  fontFamily: "Roboto", "Helvetica", "Arial", sans-serif
  fontWeight: 300
  fontSize: "6rem"
  lineHeight: 1.167
  letterSpacing: "-0.01562em"
> h2: Object
> h3: Object
> h4: Object
> h5: Object
> h6: Object
> subtitle1: Object
> subtitle2: Object
> body1: Object
> body2: Object
> button: Object
> caption: Object
> overline: Object
```

variant

ישנים ערכים דיפולטיביים של גודל טקסט גובהה תו והגדרות נספנות לטקסט המומלצים לעבודה באפליקציות המבוססות וUI.

על מנת להשתמש בעיצוב הדיפולטיבי נוסף לקומפוננט Typography מאפיין בשם variant ונדיר את העיצוב בערך של המאפיין זהה.

- הוספנו לקומפוננט Typography את המאפיין variant וקבענו את ערכי העיצוב שלו לערך הדיפולטיבי של 4h
- בדף זה בדוקומנטציה של UIW ניתן לראות את כל הערכים הדיפולטיביים לעיצוב טקסט
- אם נלחץ על אייקון החץ בתחילת שם העיצוב יוכל לראות ממה הרכיבו את הערך הדיפולטיבי

לרשימת הערכים דיפולטיביים לטקסט בlienק הבא
[https://mui.com/material-ui/customization/default-theme/?expand-path=\\$.typography#explore](https://mui.com/material-ui/customization/default-theme/?expand-path=$.typography#explore)

component attribute

ניתן להוסיף מאפיין נוסף בשם component ובכך לנתק את העיצוב של הטקסט מאלמנט ה - HTML

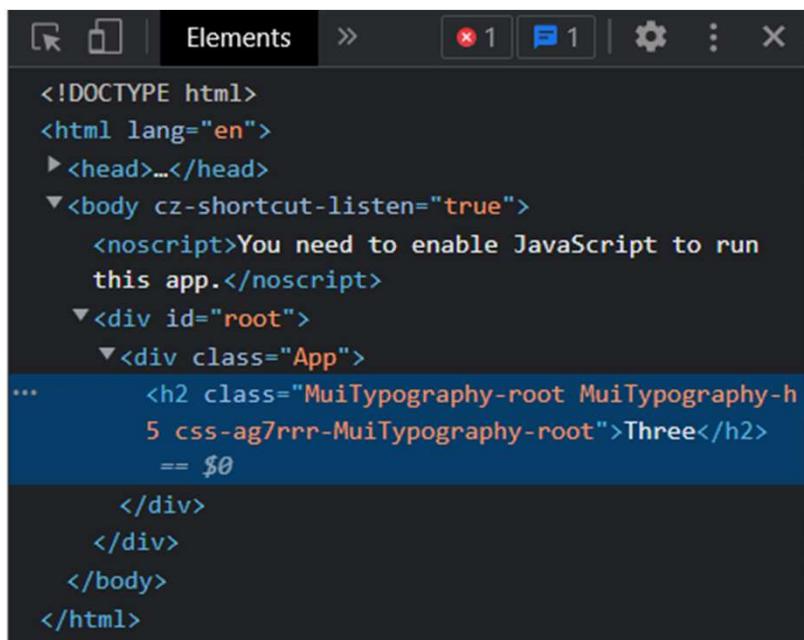
דוגמה של להלן:

- נקבע כי העיצוב של הטקסט יהיה לפי הערך הדיפולטיבי שモופיע במאפיין ה - variant בלוור h5
- נקבע כי אלמנט ה - HTML יהיה h2
- התוצאה בדפדף:

▪ ניתן לראות כי גודל הטקסט תואם לעיצוב של h5 אולם אלמנט ה - HTML הוא למעשה h2

```
12 const TypographyComponent = () => {  
13     return (  
14         <Typography component="h2" variant="h5">  
15             Three  
16         </Typography>  
17     );  
18 };
```

Three



Font width

- בעזרה הוספה המאפיין `fontWeight` נוכל לקבוע את עובי הטקסט כפי ש谟ופיע בדוגמה שלහן

- התוצאה בדף

```
20 const TypographyComponent = () => {
21   return (
22     <>
23       <Typography fontWeight={300}>Light</Typography>
24       <Typography fontWeight={400}>Regular</Typography>
25       <Typography fontWeight={500}>Medium</Typography>
26       <Typography fontWeight={700}>Bold</Typography>
27     </>
28   );
29 }
```

Light
Regular
Medium
Bold

! באותו הלינק שהציג בשקפים הקודמים ניתן להגיע לפרמטרים הללו ב - **Typography**

Font colors

מאפיין נוסף שBITON להוסיף לקומפוננט Typography הוא המאפיין color שייעצב את הטקסט בהתאם לצבע שנקבע

- בערכים הדיפולטיים כמעט לכל צבע יש שלוש אופציות של תצוגה:

- light – מצב תצוגה בהיר
- main – ערכים דיפולטיים
- dark – מצב תצוגה בהה

- לצבע grey אין את מצביו התצוגה הלו

```
31 const TypographyComponent = () => {
32   return (
33     <>
34       <Typography color="primary.light">Light</Typography>
35       <Typography color="primary.main">Main</Typography>
36       <Typography color="primary.dark">Dark</Typography>
37       <Typography color="secondary">secondary</Typography>
38       <Typography color="error">error</Typography>
39       <Typography color="warning.main">warning</Typography>
40       <Typography color="info.main">info</Typography>
41       <Typography color="success.main">success</Typography>
42       <Typography color="grey">grey</Typography>
43     </>
44   );
45 };
```

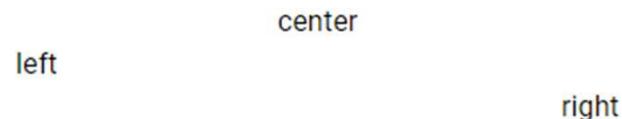
Light
Main
Dark
secondary
error
warning
info
success
grey

! ניתן לדוחס את הערכים הדיפולטיים הללו עם צבעים משלמו

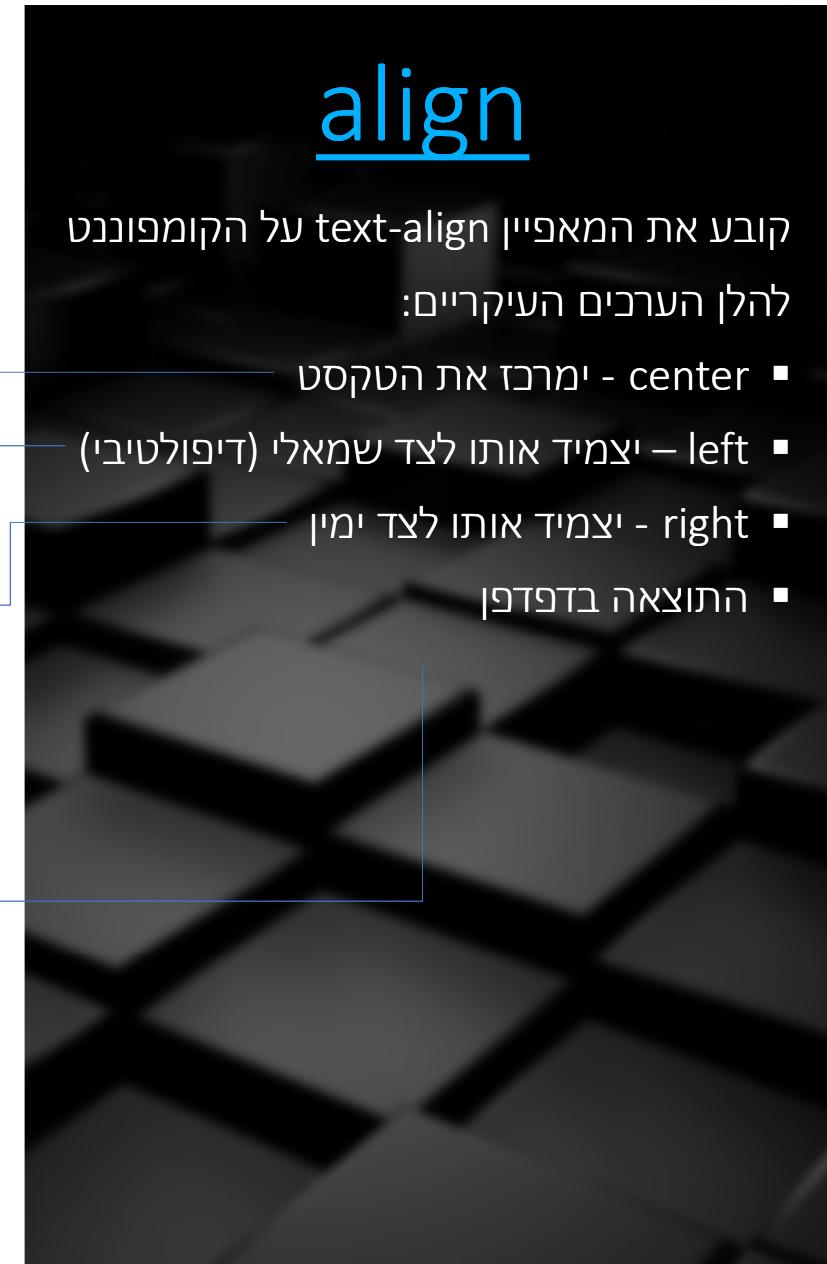
align

קובע את המאפיין `text-align` על הקומפוננט
להלן הערכים העיקריים:

```
49 const TypographyComponent = () => {  
50   return (  
51     <>  
52     <Typography align="center">center</Typography>  
53     <Typography align="left">left</Typography>  
54     <Typography align="right">right</Typography>  
55   </>  
56 );  
57};
```



<https://mui.com/material-ui/api/typography/#props>



```
60 const TypographyComponent = () => {
61   return (
62     <>
63       <Typography>
64         Lorem ipsum dolor sit amet, consectetur adipisicing
65           elit. Dolores,
66           architecto. Repellat doloremque maxime minima est
67             temporibus, id quo, ea
68             aliquid quae earum culpa dolor aut porro ipsum
69               numquam cupiditate fugit!
70     </Typography>
71     <hr />
72     <Typography nowrap>
73       Lorem ipsum dolor sit amet consectetur adipisicing
74         elit. Explicabo velit
75           molestiae exercitationem voluptatem quo labore
76             voluptate, incident
77             deserunt commodi necessitatibus. Dolore consectetur
78               necessitatibus
79                 laudantium.
80     </Typography>
81   </>
82 );
83 };
```

noWrap

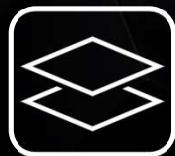
- מאפיין שאחרי על הורדת שורה כאשר הטקסט מגע לסוף האלמנט שבו הוא נמצא
- אם לא נוסיף את המאפיין `Wrap` כאשר הטקסט יגיע לסוף האלמנט שמכיל אותו הוא ירד שורה
- הוספת המאפיין `Wrap` תוביל לכך שהtekst לא ירד שורה אלא יהיה שלוש נקודות במקומות בו הטקסט יהיה אמר לודת שורה

התוצאה בדף

- הטקסט בקומפוננט הראשון יורד שורה ברגע שאין לו מקום במילוי
- הטקסט בקומפוננט השני לא יורד שורה ובמקום מסמן בעזרת שלוש נקודות שהtekst ממשיר

Lorem ipsum dolor sit amet, consectetur adipisicing elit.
Dolores, architecto. Repellat doloremque maxime minima
est temporibus, id quo, ea aliquid quae earum culpa dolor
aut porro ipsum numquam cupiditate fugit!

Lorem ipsum dolor sit amet consectetur adipisicing elit. ...



SX

“All system properties are available via the sx prop. In addition, the sx prop allows you to specify any other CSS rules you may need”

[mui.com](https://mui.com/material-ui/react-box/#the-sx-prop)

<https://mui.com/material-ui/react-box/#the-sx-prop>



```
80  const TypographyComponent = () => {
81    return (
82      <Typography
83        sx={{←
84          margin: 2,
85          backgroundColor: "warning.main",
86          padding: 2,
87          textAlign: "center",
88        }}>
89        SX ←
90      </Typography>
91    );
92  };
```

SX prop

מופיעין המקבל אובייקט קונפיגורציות של `css` ומאפשר שינוי של ה-`css` בкомпонент

- הוספנו את המאפיין `sx` לkomponennt `Typography`

▪ המאפיין מקבל אובייקט קונפיגורציות עם כל מאפייני ה-`CSS` שאנו מעוניינים לשנות

- הטקסט של הקומponent

- התוצאה בדף:

SX

<https://mui.com/system/getting-started/the-sx-prop/>

!-link להסביר נוסף על המאפיין SX



Divider

A divider is a thin line that groups content in lists and layouts.

<https://mui.com/material-ui/api/divider/>



Regular & light

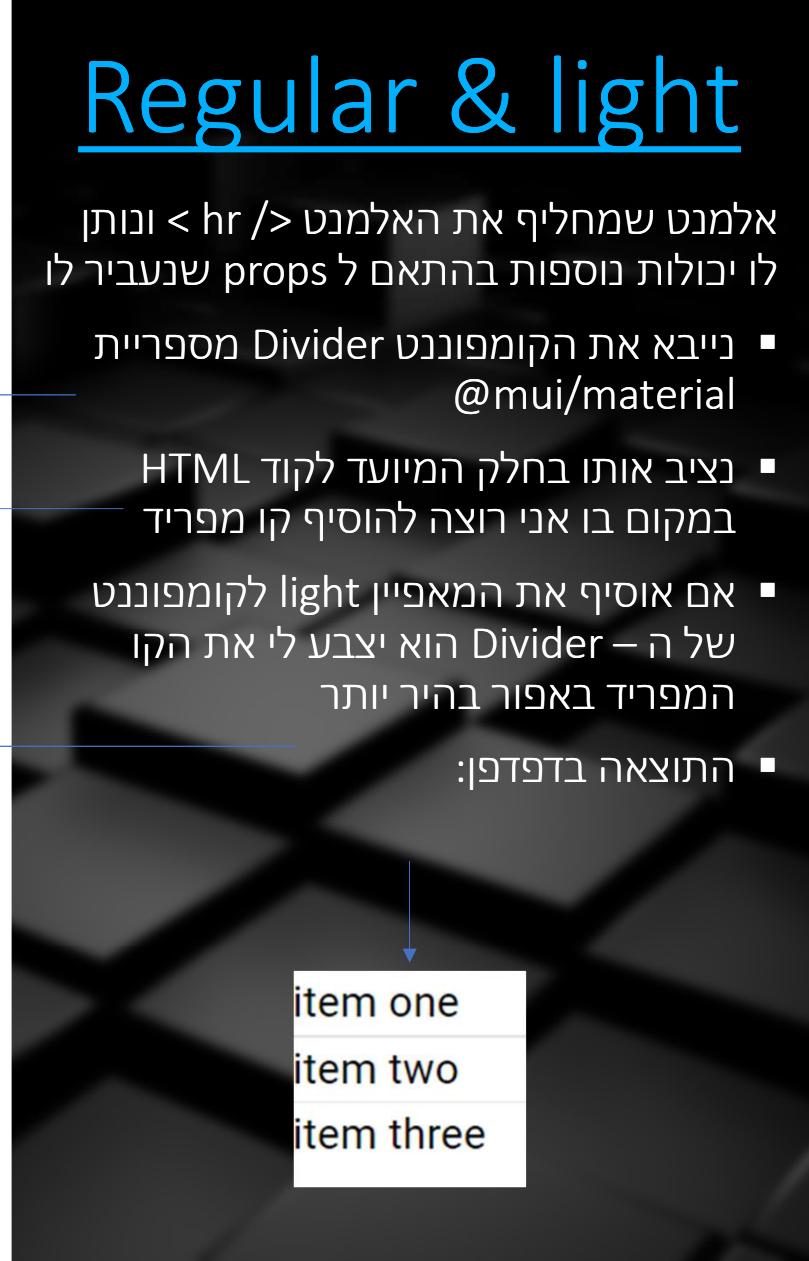
אלמנט שמחליף את האלמנט `</hr>` ונותן
לו יכולות נוספות בהתאם לprops שנעביר לו

- ניבא את הקומponent Divider מספרית
`@mui/material`

▪ נציב אותו בחלק המועד לקוד HTML
במוקם בו אני רוצה להוסיף קו מפרד

- אם אוסיף את המאפיין `light` לקומponent
של ה-`Divider` הוא יציבו לי את הקוו
המפרד באפור בהיר יותר

▪ התוצאה בדף:



item one
item two
item three

DividerComponent.jsx

```
material-ui-sandbox > src > components > DividerComponent.jsx > ...
1 import React from "react";
2 import Typography from "@mui/material/Typography";
3 import Divider from "@mui/material/Divider"; ←
4
5 const DividerComponent = () => {
6   return (
7     <>
8       <Typography>item one</Typography>
9       <Divider /> ←
10      <Typography>item two</Typography>
11      <Divider light /> ←
12      <Typography>item three</Typography>
13     </>
14   );
15 }
16
17 export default DividerComponent;
```

Vertical Divider

על מנת ליצור מפריד אנחנו צריכים לבצע את הפעולות הבאות

- המוביל שיכיל בתוכו את האלמנטים והמפריד צריך להיות על "display="flex"
- Divider – צריך להוסיף שני props לkomponennt:
 - orientation – להגדיר אותו לאנכי
 - flexItem – מגדר את האלמנט תחת הכללים של flex
- התוצאה בדף

```
19 const DividerComponent = () => {
20   return (
21     <Box display="flex">
22       <Typography p={1}>item one</Typography>
23       <Divider orientation="vertical" flexItem />
24       <Typography p={1}>item two</Typography>
25     </Box>
26   );
27 }
```

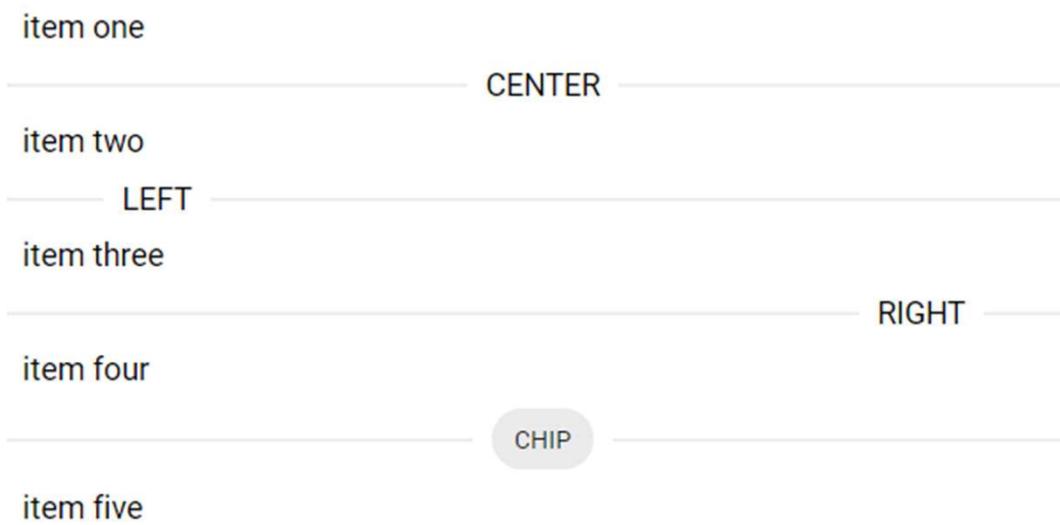
item one | item two

textAlign

בעזרת prop זה נוכל לשים בתוך הkn
המפריד טקסט

- על מנת למרכז את הטקסט משתמש גם
בתגית הפתוחת של ה – Divider ונציב את
הטקסט בית התגית הפתוחת לתגית
הסוגרת
- אם נרצה להציג את הטקסט לצד שמאל
נשימוש ב textAlign עם הערך left
- אם נרצה להציג את הטקסט לצד ימין
נשימוש ב textAlign עם הערך right
- מלבד טקסט נוכל להכניס בקומפוננט
Divider גם קומפוננטות אחרות בגין
שימושו את הטקסט שאעביר לפופ
label בעיצוב דיפולטיבי של הקומפוננט

```
const DividerComponent = () => {
  return (
    <>
      <Typography p={1}>item one</Typography>
      <Divider>CENTER</Divider> ←
      <Typography p={1}>item two</Typography>
      <Divider textAlign="left">LEFT</Divider> ←
      <Typography p={1}>item three</Typography>
      <Divider textAlign="right">RIGHT</Divider>
      <Typography p={1}>item four</Typography>
      <Divider>
        <Chip label="CHIP" /> ←
      </Divider>
      <Typography p={1}>item five</Typography>
    </>
  );
};
```



התוצאה בדף

ניתן לראות כי במקרים המפרידים נוספים
טקסטים במקומות שקבענו להם



Button

Buttons allow users to take actions, and make choices, with a single tap.

<https://mui.com/material-ui/api/button/>



variant & disabled

הprop `variant` מתייחס לעיצוב הכללי של אלמנט הcptor בדוגמה שלהן:

- `text` – נראה רק את הטקסט ואם נעלם העבר על אזור הcptor הרקע יהיה כהה יותר והמצבי "הפרק מבחן לא צבע רקע
- `contained` - ימלא את הcptor בצבע רקע
- `outlined` - הcptor יהיה שקוף ורק המסגרת שלו תהיה צבעה
- `disabled` – לא יהיה ניתן ללחוץ על הcptor והוא יצביע באפור
- התוצאה בדף

ButtonComponent.jsx

```
src > components > ButtonComponent.jsx > ButtonComponent.js  
1 import { Button } from "@mui/material";  
2 import { Box } from "@mui/system";  
3 import React from "react";  
4  
5 const ButtonComponent = () => {  
6   return (  
7     <Box m={2} display="flex" justifyContent="space-between">  
8       <Button variant="text">Text</Button> ←  
9       <Button variant="contained">Contained</Button> ←  
10      <Button variant="outlined">Outlined</Button> ←  
11      <Button variant="contained" disabled> ←  
12        disabled  
13      </Button>  
14    </Box>  
15  );  
16};  
17  
18 export default ButtonComponent;
```

TEXT

CONTAINED

OUTLINED

DISABLED

```
20 const ButtonComponent = () => {
21   return (
22     <Box m={2} display="flex" justifyContent="space-between">
23       <Button variant="contained" color="primary">
24         primary
25       </Button>
26       <Button variant="contained" color="secondary">
27         secondary
28       </Button>
29       <Button variant="contained" color="success">
30         success
31       </Button>
32       <Button variant="contained" color="error">
33         error
34       </Button>
35       <Button variant="contained" color="info">
36         info
37       </Button>
38       <Button variant="contained" color="warning">
39         warning
40       </Button>
41     </Box>
42   );
43 };
```

color

מופיע שיקב את הצבע של הכפתור

- primary - הצבע הדיפולטיבי שהוא כחול בהה
- secondary - סגול
- success - ירוק
- error - אדום
- info - תכלת
- warning - כתום

התוצאה בדף:

PRIMARY

SECONDARY

SUCCESS

ERROR

INFO

WARNING

```
46 const ButtonComponent = () => {
47   return (
48     <Box sx={{ "& button": { m: 1 } }}>
49       <div>
50         <Button variant="outlined" size="small">
51           Small
52         </Button>
53         <Button variant="outlined" size="medium">
54           Medium
55         </Button>
56         <Button variant="outlined" size="large">
57           Large
58         </Button>
59       </div>
60     </Box>
61   );
62 };
```

SMALL MEDIUM LARGE



size

מאפיין שיקבע את גודל הכפתור

- ישנם שלושה אופציות לגודל כפתור:

- small - הגודל הקטן ביותר

- medium – הגודל הדיפולטיבי

- large - הגודל הגדול ביותר

- התוצאה בדף:

Button with Icon

ניתן לקבוע היקן וציב איקון במידה ואנו מעוניינים שהוא יציג על הכיתוב בכפטור בעזרת המאפיינים הבאים:

- `startIcon` - הצבת האיקון לפני הכיתוב על הכפטור
- `endIcon` – הצבת האיקון אחרי הכיתוב על הכפטור
- התוצאה בדף:

```
69 const ButtonComponent = () => {
70   return (
71     <>
72       <Box sx={{ "& button": { m: 1 } }} justifyContent="center" display="flex">
73         <Button variant="outlined" startIcon={<DeleteIcon />}>
74           Delete
75         </Button>
76         <Button variant="contained" endIcon={<SendIcon />}>
77           Send
78         </Button>
79       </Box>
80
81       <Box>
82         <Button variant="contained" endIcon={<SendIcon />} fullWidth>
83           Send
84         </Button>
85       </Box>
86     </>
87   );
88 };
```



```
83 const ButtonComponent = () => {
84   return (
85     <Box sx={{ "& button": { m: 1 } }}>
86       <IconButton aria-label="delete"> ←
87         <DeleteIcon /> ←
88       </IconButton>
89       <IconButton aria-label="delete" disabled> ←
90         <DeleteIcon /> ←
91       </IconButton>
92       <IconButton color="secondary"> ←
93         <AlarmIcon /> ←
94       </IconButton>
95       <IconButton color="primary"> ←
96         <AddShoppingCartIcon fontSize="large" /> ←
97       </IconButton>
98     </Box>
99   );
100};
```



IconButton

קומponeנט שיציג כפתורים כאיקונים ויאפשר האזנה לארוחים

- הקומponeנט IconButton משמש בambil לkomponeent של איקון

- רצוי להוסיף את המאפיין aria-label על מנת לתת שם לבפתור

- ניתן להוסיף את המאפיין disabled ואם הערך שלו הוא true הכפתור יהיה מונטראל

- התוצאה בדף:



Cards

Cards contain content and actions about a single subject and are entry points to more complex and detailed information

<https://mui.com/material-ui/api/card/>



Card Header

יצירת בותחות בראש הכרטיס

- בתוך הקומפוננט Card של UI שתשמש כמייל לכל הכרטיס
- נציב את הקומפוננט CardHeader שיבול לקביל את המאפיינים:
 - – בותחת משנה subheader
 - – בותחת ראשית title

CardComponent.jsx

```
src > components > CardComponent.jsx > ...
1  import React from "react";
2  import Card from "@mui/material/Card";
3  import CardHeader from "@mui/material/CardHeader";
4
5  const CardComponent = () => {
6    return (
7      <Card sx={{ width: 250, m: 2 }}> ←
8        <CardHeader title="Card Header" subheader="Card Subheader" />
9      </Card>
10     );
11   };
12
13 export default CardComponent;
```

Card Header

Card Subheader

<https://mui.com/material-ui/api/card-header>

התוצאה בדף

```
30 const CardComponent = () => {
31   return (
32     <Card sx={{ width: 250, m: 2 }} square raised>
33       <CardHeader title="Card Header" />
34     </Card>
35   );
36 };
```

Card Header

square & raised

מאפיינים על הקומponent Card

- square - יփוך את גבולות הברטיס מעוגלים לזרויים
- raised – יגדיל את הצל של הברטיס

התוצאה בדף

```
30 const CardComponent = () => {
31   return (
32     <Card sx={{ width: 250, m: 2 }} variant="outlined">
33       <CardHeader title="Card Header" />
34     </Card>
35   );
36 };
```

Card Header



```
39 const CardComponent = () => {
40   return (
41     <Card sx={{ width: 250, m: 2 }} raised>
42       <CardHeader title="Card Header" />
43       <CardContent><!--
44         <Typography>Card content</Typography>
45       </CardContent>
46     </Card>
47   );
48 };
```

Card Header

Card content

CardContent

- באזור זה יוצג התוכן המרכזי של הברטיס. בדוק כלל הוא יכול מידע דינامي
- ניצור אזור תוכן לברטיס על ידי הוספת **CardContent**
 - נציב בתוכו את הקומפוננט **Typography** עם MERCHANTABILITY התווים

התוצאה בדף

! קישור ל – **:CardContent API**
<https://mui.com/material-ui/api/card-content/>

CardActionArea

אזור שקומponent זה יعطוף ויקבל אינטראקציה כאשר יעלו עם העכבר על האלמנט והוא ישנה את צבעו ובמקומ חץ יופיע cursor של pointer

- נعطוף את הקומפוננטות CardContent בקומponent CardActionArea על מנת לסמן לגולש כי ניתן ללחוץ על אזור זה ולקבל: פרטיים נוספים, לעבור לקומponent אחר או כל פעולה אחרת שנגדי

```
51 const CardComponent = () => {
52   return (
53     <Card sx={{ width: 250, m: 2 }} raised>
54       <CardActionArea>
55         <CardHeader title="Card Header" />
56         <CardContent>
57           <Typography>Card content</Typography>
58         </CardContent>
59       </CardActionArea>
60     </Card>
61   );
62 }
```

Card Header

Card content

התוצאה בדף

! קישור ל API :CardActionArea
<https://mui.com/material-ui/api/card-action-area/>

CardMedia

על מנת להוסיף תמונה, סאונד או יideo בכרטיס נשתמש בקומפוננט מיוחד לצורך כר

- נציב את הקומפוננט CardMedia בתוך הקומפוננט Card Component – נקבע כי האלמנט HTML יהיה img – נקבע גובה אחיד לכל התמונות של הכרטיסים כדי לשמר על אחידות סגנונית
- image – המקבילה ל – src באלמנט img מאפיין זה יקבל כתובות URL בו נמצאת המדיה המבוקשת
- alt – טקסט חלופי במידה וכתובות ה – URL לא מובילה לתמונה בלבד

התוצאה בדף

! קישור ל API :CardMedia

<https://mui.com/material-ui/api/card-media/>

```
66 const CardComponent = () => {  
67   return (  
68     <Card sx={{ width: 250, m: 2 }} raised>  
69       <CardMedia component="img" height="140"  
70         image="/assets/images/bird.jpg" />  
71     </Card>  
72   );  
73 };
```

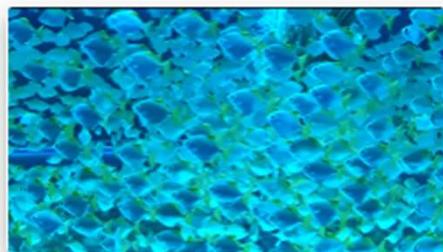


CardMedia

ניתן להשתמש באותה קומפוננט גם כדי להציג וידאו

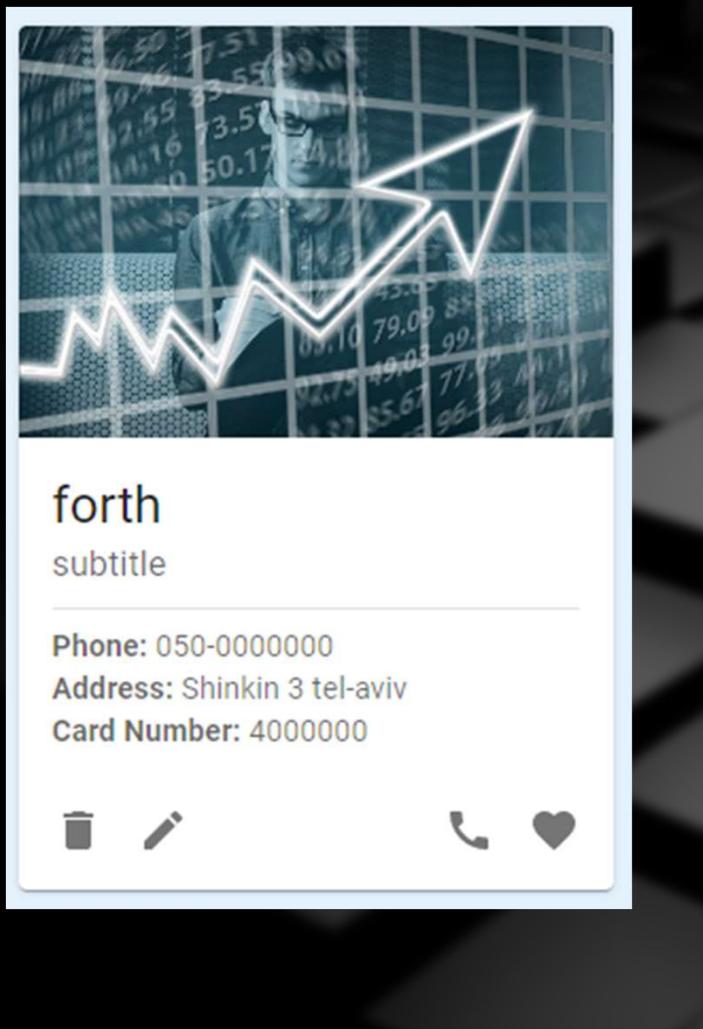
- המאפיינים של קומפוננט זה בדוגמה הבאה:
 - component - אלמנט ה – video בירינה יהיה אלמנט video
 - src - כתובות ה – URL שלקובץ הוידאו
 - autoplay – ניגון אוטומטי של הוידאו עם יצירה הקומפוננט
 - loop - חזרה על הוידאו כשהוא מסתiem באופן אוטומטי
 - muted – השתקת הסאונד של הוידאו

```
75 const CardComponent = () => {  
76   return (  
77     <Card sx={{ width: 250, m: 2 }} raised>  
78       <CardMedia  
79         component="video"  
80         height="140"  
81         src="/assets/videos/Fish.mp4"  
82         autoComplete  
83         loop  
84         muted  
85       />  
86     </Card>  
87   );  
88};
```



התוצאה בדף

משימת Card



Business-cards-app

- יבא את הספריות הקשורות ל UI material לפרויקט כפי שモופיע בשקפים הקודמים
- יבא את ספריית Roboto לפרויקט
- עורך בדיקה בקובץjs.App שהספרייהעובדות
- צור את הnetteib axjs.Card.js – צור קומפוננט מסווג פונקציה שבעזרת ספריית UI material שתיצור אלמנט HTML של כרטיס ביקור לעסק שנראה כמו בדוגמה משמאל.
- הצב את הקומפוננט שיצרת בתוך הקומפוננט js.App
- הפעיל את השרת של React ובודק שאכן אתה רואה את הברטיס כפי שモופיע בדוגמה
- פתח את הקונסול בדףן וודא שאין בו שגיאות או הערות אחרות



List

Lists are continuous, vertical indexes of text or images. They are composed of items containing primary and supplemental actions, which are represented by icons and text.

<https://mui.com/material-ui/react-list/>



List & ListItem

ישנים.

▪ הוסףנו

ListComponent.jsx

```
src > components > ListComponent.jsx > ...
1  import List from "@mui/material/List";
2  import ListItem from "@mui/material/ListItem";
3  importListItemText from "@mui/material/ListItemText";
4
5  const ListComponent = () => {
6    return (
7      <List>
8        <ListItem disablePadding>
9          <ListItemText primary="Inbox" />
10         </ListItem>
11        <ListItem disablePadding>
12          <ListItemText primary="Drafts" />
13        </ListItem>
14      </List>
15    );
16  };
17
18  export default ListComponent;
```

Inbox

Drafts

15/11/2022

לרשימת ה **ערכים דיפולטיביים** לטעסט בlienק הבא

<https://mui.com/material-ui/api/list/>

disablePadding

```
27 const ListComponent = () => {
28   return (
29     <List disablePadding>
30       <ListItem disablePadding>
31         <ListItemText primary="Inbox" />
32       </ListItem>
33       <ListItem disablePadding>
34         <ListItemText primary="Drafts" secondary="15/11/2022" />
35       </ListItem>
36     </List>
37   );
38 };
```

Inbox
Drafts
15/11/2022

ישנים ערכים דיפולטיביים של גודל טקסט גובהה תו והגדרות נוספות לטקסט המומלצים לעבודה באפליקציות המבוססות וUI.

על מנת להשתמש בעיצוב הדיפולטיבי נוסיף לקומפוננט Typography מאפיין בשם variant ונגדיר את העיצוב בערך של המאפיין זהה.

- הוספנו לקומפוננט Typography את המאפיין variant וקבענו את ערכי העיצוב שלו לערך הדיפולטיבי של 4h

- בדף זה בדוקו מנכזיה של וUI ניתן לראות את כל הערכים הדיפולטיביים לעיצוב טקסט

- אם נלחץ על אייקון החץ בתחילת שם העיצוב יוכל לראות ממה הרכיבו את הערך הדיפולטיבי

!רשימת ה ערכים דיפולטיביים לטקסט בlienck הבא
<https://mui.com/material-ui/api/list/>

dense

```
41 const ListComponent = () => {
42   return (
43     <List dense>
44       <ListItem>
45         <ListItemText primary="Inbox" />
46       </ListItem>
47       <ListItem>
48         <ListItemText primary="Drafts" secondary="15/11/2022" />
49       </ListItem>
50     </List>
51   );
52 };
```

Inbox
Drafts
15/11/2022

ישנים ערכים דיפולטיביים של גודל טקסט גבוהה تو והגדרות נוספות לטקסט המומלצים לעבודה באפליקציות המבוססות וUI.

על מנת להשתמש בעיצוב הדיפולטיבי נוסף לקומפוננט Typography מאפיין בשם variant ונדיר את העיצוב בערך של המאפיין זהה.

- הוספנו לקומפוננט Typography את המאפיין variant וקבענו את ערכי העיצוב שלו לערך הדיפולטיבי של 4 h4

- בדף זה בדוקו מנציה של וUI ניתן לראות את כל הערכים הדיפולטיביים לעיצוב טקסט

- אם נלחץ על אייקון החץ בתחילת שם העיצוב יוכל לראות ממה הרכיבו את הערך הדיפולטיבי

! **רשימת ה ערכים דיפולטיביים לטקסט בリンק הבא**
<https://mui.com/material-ui/api/list/>

ListSubheader

```
55 const ListComponent = () => {  
56   return (  
57     <List>  
58       <ListSubheader>Sub header</ListSubheader>  
59       <ListItem>  
60         <ListItemText primary="Inbox" />  
61       </ListItem>  
62       <ListItem>  
63         <ListItemText primary="Drafts" />  
64       </ListItem>  
65     </List>  
66   );  
67 };
```

Sub header

Inbox

Drafts

ישנים ערכים דיפולטיביים של גודל טקסט
גובהה תו והגדרות נוספות לטקסט המומלצים
לעבודה באפליקציות המבוססות וUI.

על מנת להשתמש בעיצוב הדיפולטיבי נוסף
לקומponent Typography מאפיין בשם
variant ונדיר את העיצוב בערך של המאפיין
זהה.

- הוספנו לקומponent Typography את
המאפיין variant וקבענו את ערכי העיצוב
שלו לערך הדיפולטיבי של 4h
- בדף זה בדוקומנטציה של וUI ניתן לראות
את כל הערכים הדיפולטיביים לעיצוב
טקסט

- אם נלחץ על אייקון החץ בתחילת שם העיצוב
ונכל לזרות ממנו הרכיבו את הערך הדיפולטיבי

לרשימת הרכים דיפולטיביים לטקסט בlienck הבא

<https://mui.com/material-ui/api/list/>

divider

ישנים ערכים דיפולטיביים של גודל טקסט גובההתו והגדרות נוספות לטקסט המומלצים לעבודה באפליקציות המבוססות וUI.

על מנת להשתמש בעיצוב הדיפולטיבי נוסף לקומפוננט Typography מאפיין בשם variant ונדיר את העיצוב בערך של המאפיין זהה.

- הוספנו לקומפוננט Typography את המאפיין variant וקבענו את ערכי העיצוב שלו לערך הדיפולטיבי של 4h
- בדף זה בדוקו מנגנזה של וUI ניתן לראות את כל הערכים הדיפולטיביים לעיצוב טקסט

- אם נלחץ על איקון החץ בתחילת שם העיצוב יוכל לראות ממנו הרכיבו את הערך הדיפולטיבי

! **רשימת הערךדים דיפולטיביים לטקסט בリンק הבא**
<https://mui.com/material-ui/api/list/>

```
70 const ListComponent = () => {
71   return (
72     <List>
73       <ListItem divider>
74         <ListItemText primary="Inbox" />
75       </ListItem>
76       <ListItem divider>
77         <ListItemText primary="Drafts" />
78       </ListItem>
79     </List>
80   );
81 };
```

Inbox

Drafts

ListButtonItem

ישנם ערכים דיפולטיביים של גודל טקסט גובהה תו והגדרות נוספות לטקסט המומלצים לעובדה באפליקציות המבוססות וUI.

על מנת להשתמש בעיצוב הדיפולטיבי נוסף לקומפוננט Typography מאפיין בשם variant ונדיר את העיצוב בערך של המאפיין זהה.

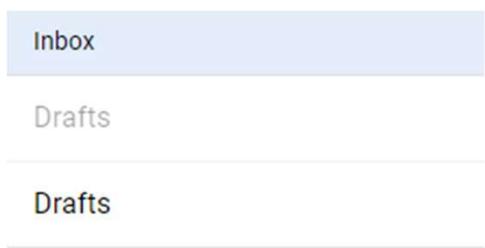
- הוספנו לקומפוננט Typography את המאפיין variant וקבענו את ערכי העיצוב שלו לערך הדיפולטיבי של 4h

- בדף זה בדוקומנטציה של וUI ניתן לראות את כל הערכים הדיפולטיביים לעיצוב טקסט

- אם נלחץ על אייקון החץ בתחילת שם העיצוב יוכל לראות ממה הרכיבו את הערך הדיפולטיבי

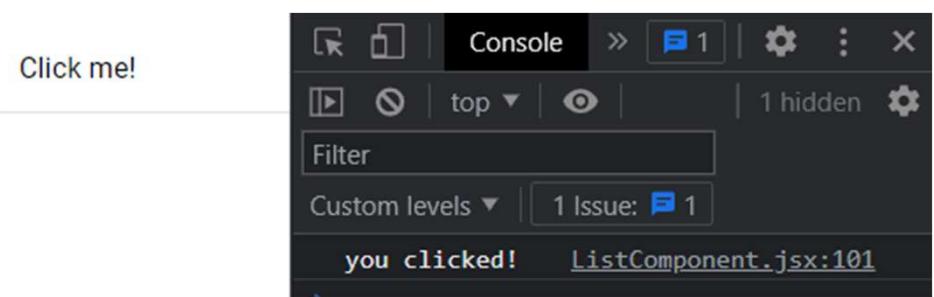
לרשימת הערך דיפולטיביים לטקסט בlienck הבא
<https://mui.com/material-ui/api/list/>

```
84 const ListComponent = () => {
85   return (
86     <List>
87       <ListItemButton divider selected dense>
88         <ListItemText primary="Inbox" />
89       </ListItemButton>
90       <ListItemButton divider disabled>
91         <ListItemText primary="Drafts" />
92       </ListItemButton>
93       <ListItemButton divider component="a" href="#simple-list">
94         <ListItemText primary="Drafts" />
95       </ListItemButton>
96     </List>
97   );
98 }
```



Passing Events

```
101 const onClick = () => console.log("you clicked!");
102
103 const ListComponent = () => {
104   return (
105     <List>
106       <ListItemButton divider onClick={onClick}>
107         <ListItemText primary="Click me!" />
108       </ListItemButton>
109     </List>
110   );
111 };
```



ישנים ערכים דיפולטיביים של גודל טקסט גובהה תו והגדרות נוספות לטקסט המומלצים לעבודה באפליקציות המבוססות וUI.

על מנת להשתמש בעיצוב הדיפולטיבי נוסף לקומponent Typography מאפיין בשם variant ונדיר את העיצוב בערך של המאפיין זהה.

- הוספנו לקומponent Typography את המאפיין variant וקבענו את ערכי העיצוב שלו לערך הדיפולטיבי של 4h

- בדף זה בדוקומנטציה של UIW ניתן לראות את כל הערכים הדיפולטיביים לעיצוב טקסט

- אם נלחץ על אייקון החץ בתחילת שם העיצוב יוכל לראות ממה הרכיבו את הערך הדיפולטיבי

רשימת הערכים דיפולטיביים לטקסט בקישור הבא
<https://mui.com/material-ui/api/list/>

ListIcon

```
114 const ListComponent = () => {  
115   return (  
116     <List>  
117       <ListItemButton divider dense>  
118         <ListItemIcon>  
119           <DraftsIcon />  
120         </ListItemIcon>  
121         <ListItemText primary="Inbox" />  
122       </ListItemButton>  
123     </List>  
124   );  
125};
```



ישנים ערכים דיפולטיבים של גודל טקסט גובהה תו והגדרות נוספות לטקסט המומלצים לעובדה באפליקציות המבוססות וUI.

על מנת להשתמש בעיצוב הדיפולטיבי נוסף לקומפוננט Typography מאפיין בשם variant ונדיר את העיצוב בערך של המאפיין זהה.

- הוספנו לקומפוננט Typography את המאפיין variant וקבענו את ערכי העיצוב שלו לערך הדיפולטיבי של `h4`
- בדף זה בדוקומנטציה של וUI ניתן לראות את כל הערכים הדיפולטיבים לעיצוב טקסט
- אם נלחץ על אייקון החץ בתחילת שם העיצוב יוכל לראות ממה הרכיבו את הערך הדיפולטיבי

רשימת הרכים דיפולטיבים לטקסט בlienck הבא
<https://mui.com/material-ui/api/list/>

ListAvatar

ישנם ערכים דיפולטיביים של גודל טקסט גובהה תו והגדרות נוספות לטקסט המומלצים לעבודה באפליקציות המבוססות וUI.

על מנת להשתמש בעיצוב הדיפולטיבי נסיף לקומפוננט Typography מאפיין בשם variant ונדיר את העיצוב בערך של המאפיין זהה.

- הוספנו לקומפוננט Typography את המאפיין variant וקבענו את ערכי העיצוב שלו לערך הדיפולטיבי של 4h

- בדף זה בדוקו מנגנזה של וUI ניתן לראות את כל הערכים הדיפולטיביים לעיצוב טקסט

- אם נלחץ על איקון החץ בתחילת שם העיצוב יוכל לראות ממנו הרכיבו את הערך הדיפולטיבי

! **רשימת הערכים דיפולטיביים לטקסט בlienck הבא**
<https://mui.com/material-ui/api/list/>

```
128 const ListComponent = () => {
129   return (
130     <List>
131       <ListItemButton divider>
132         <ListItemIconAvatar>
133           <Avatar src="https://cdn.pixabay.com/photo/2022/
134             10/24/14/41/blackbird-7543630_960_720.jpg" />
135           <ListItemText primary="Avatar" />
136         </ListItemIcon>
137       </ListItemButton>
138     );
139   };

```



Avatar



Dialog

“Dialogs inform users about a task and can contain critical information, require decisions, or involve multiple tasks.”

mui.com

<https://mui.com/material-ui/api/dialog/>



ListItemAvatar

ישנם ערכים דיפולטיביים של גודל טקסט גובהה תו והגדרות נספנות לטקסט המומלצים לעובודה באפליקציות המבוססות וՄ.

על מנת להשתמש בעיצוב הדיפולטיבי נוסף לקומponent Typography מאפיין בשם variant ונדיר את העיצוב בערך של המאפיין זהה.

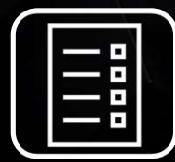
- הוספנו לקומponent Typography את המאפיין variant וקבענו את ערכי העיצוב שלו לערך הדיפולטיבי של 4h

- בדף זה בדוקו מנצריה של וՄ ניתן לראות את כל הערכים הדיפולטיביים לעיצוב טקסט

- אם נלחץ על אייקון החץ בתחילת שם העיצוב יוכל לראות ממה הרכיבו את הערך הדיפולטיבי

!
רשימת הערךים דיפולטיביים לטקסט בlienק הבא
<https://mui.com/material-ui/api/list/>

```
client > src > sandbox > mui-sandbox > Dialog > MuiDialog.jsx > MuiDialog
1 import * as React from "react";
2 import Button from "@mui/material/Button";
3 import Dialog from "@mui/material/Dialog";
4 import DialogActions from "@mui/material/DialogActions";
5 import DialogContent from "@mui/material/DialogContent";
6 import DialogContentText from "@mui/material/DialogContentText";
7 importDialogTitle from "@mui/material/DialogTitle";
8
9 const MuiDialog = () => [
10   const [open, setOpen] = React.useState(false);
11   const handleClickOpen = () => setOpen(true);
12   const handleClose = () => setOpen(false);
13
14   return (
15     <div>
16       <Button variant="outlined" onClick={handleClickOpen}>
17         Open alert dialog
18       </Button>
19
20       <Dialog
21         open={open}
22         onClose={handleClose}
23         aria-labelledby="alert-dialog-title"
24         aria-describedby="alert-dialog-description"
25         maxWidth="xs">
26         <DialogTitle id="alert-dialog-title">
27           {"Are you sure you want to delete this card?"}
28         </DialogTitle>
29         <DialogContent>
30           <DialogContentText id="alert-dialog-description">
31             This operation will delete the card from the database completely and
32             it will not be possible to retrieve the card afterwards
33           </DialogContentText>
34         </DialogContent>
35         <DialogActions>
36           <Button onClick={handleClose} color="error">
37             cancel
38           </Button>
39           <Button onClick={handleClose} autoFocus color="info">
40             Delete card
41           </Button>
42         </DialogActions>
43       </Dialog>
44     </div>
45   );
46 ]
47
48 export default MuiDialog;
```



Progress

Progress indicators commonly known as spinners, express an unspecified wait time or display the length of a process.

<https://mui.com/material-ui/react-progress/>





Alert

“An alert displays a short, important message in a way that attracts the user's attention without interrupting the user's task.”

MUI.COM

<https://mui.com/material-ui/api/snackbar/>





Snackbar

“Snackbars inform users of a process that an app has performed or will perform. They appear temporarily, towards the bottom of the screen. They shouldn't interrupt the user experience, and they don't require user input to disappear.”

MUI.COM

<https://mui.com/material-ui/api/snackbar/>



Basic Snackbar

ישנו.

▪ הוסףנו

Snackbar.jsx

```
client > src > sandbox > mui-sandbox > snackbar > Snackbar.jsx > ...
1  import MuiSnackbar from "@mui/material/Snackbar";
2  import { useState } from "react";
3  import Button from "@mui/material/Button";
4
5  const Snackbar = () => {
6    const [isOpen, setOpen] = useState(false);
7
8    return (
9      <>
10        <Button variant="outlined" onClick={() => setOpen(true)}>
11          Open Snackbar
12        </Button>
13
14        <MuiSnackbar
15          anchorOrigin={{ vertical: "top", horizontal: "right" }}
16          open={isOpen}
17          onClose={() => setOpen(prev => !prev)}
18          autoHideDuration={3000}
19          message="In Snackbar"
20        />
21      </>
22    );
23  };

```

לשימוש ה **ערכים דיפולטיביים** לטקסט בlienק הבא
<https://mui.com/material-ui/api/list/>

Layout

Layout description





Box

The Box component serves as a wrapper component for most of the CSS utility needs

<https://mui.com/material-ui/react-box/>



LayoutComp.jsx

```
src > components > LayoutComp.jsx > ...
1 import React from "react";
2 import { Box } from "@mui/material";
3
4 const LayoutComp = () => {
5   return <Box>in box</Box>;
6 }
7
8 export default LayoutComp;
```

in box

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body cz-shortcut-listen="true">
    <noscript>You need to enable JavaScript to run this app.
    </noscript>
    <div id="root">
      <div class="App">
        ... <div class="MuiBox-root css-0">in box</div> == $0
      </div>
    </div>
  </body>
</html>
```

Box

המיבל הפשוט ביותר הוא Box באופן דיפולטיבי האלמנט שי=UTF-8 יהיה <div>

בדוגמה של להלן

- אני ממחיר מהkomponentה LayoutComp את האלמנט של וענפם של Box ובתוכו כתוב התוצאה בדף
- ניתן לראות כי את הכתוב עוטף אלמנט <div>

! ניתן להוסיף לאלמנט זה את כל המאפיינים של אלמנט div כמו onClick וצדומה



Container

The container centers your content horizontally

<https://mui.com/material-ui/api/container/>



container

מיכל זה ימרכז את התוכן ומשתנה בהתאם לגודל המסך. הוא יוסיף רווח מצדדי האלמנט בגודלי מסך מסוימים

דוגמה שלהן

- אני עוטף את הטקסט באלמנט של MUI בשם container וצובע את הרקע שלו בכחול ואת הכתוב בתוכו באפור בהיר

התוצאה בדף

- ניתן לראות שהתוכן ממוקץ ויש רווח בין לבין צידי המסך

```
10 const LayoutComp = () => {
11   return (
12     <Container sx={{ backgroundColor: "blue", color: "lightgray" }}>
13       | in container
14       </Container>←
15     );
16   };

```

in container





Stack

The Stack component manages layout of immediate children along the vertical or horizontal axis with optional spacing and/or dividers between each child.

<https://mui.com/material-ui/react-stack/>



Stack

מיכל יסדר את האלמנטים הילדיים
בתוכו בסדר אורכי או רוחבי

דוגמה הראשונה

- אני עוטף את הטקסט באלמנט של MU spacing width margin וקבע בשם Stack קבועה קומפוננטות של Paper וננתן

- מציב שלוש קומפוננטות של Paper וננתן להם padding

דוגמה השנייה

- אני מוסיף את המאפיין direction וקבע אותו על row

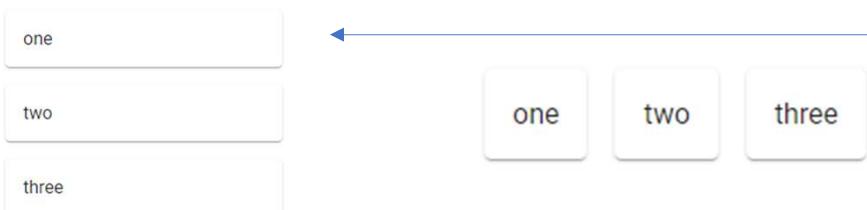
התוצאה בדף

- בתצוגה הראשונה הקומפוננטות נערמו אחד מתחת לשנייה

- בתצוגה השנייה בגלל מאפיין direction="row" הקומפוננטות מוצגות אחת ליד השנייה

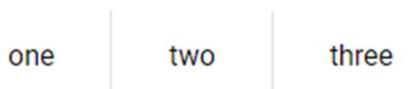
```
19 const LayoutComp = () => {  
20   return (  
21     <Stack m={2} spacing={2} width={250}>  
22       <Paper sx={{ p: 2 }}>one</Paper>  
23       <Paper sx={{ p: 2 }}>two</Paper>  
24       <Paper sx={{ p: 2 }}>three</Paper>  
25     </Stack>  
26   );  
27 };
```

```
30 const LayoutComp = () => {  
31   return (  
32     <Stack m={2} spacing={2} width={250} direction="row">  
33       <Paper sx={{ p: 2 }}>one</Paper>  
34       <Paper sx={{ p: 2 }}>two</Paper>  
35       <Paper sx={{ p: 2 }}>three</Paper>  
36     </Stack>  
37   );  
38 };
```



Stack & divider

```
42 const LayoutComp = () => {
43   return (
44     <Stack
45       m={2}
46       spacing={2}
47       width={250}
48       direction="row"
49       divider={<Divider orientation="vertical" flexItem />}>
50       <Box sx={{ p: 2 }}>one</Box>
51       <Box sx={{ p: 2 }}>two</Box>
52       <Box sx={{ p: 2 }}>three</Box>
53     </Stack>
54   );
55 };
```



יש ביכולתנו להוסיף מאפיין בשם `divider` ולהעביר לו בערך קומפוננט של `Divider` והambil `Stack` יdag לשים את החוצץ בין כל קומפוננט יلد לקומפוננט יlid אחר שהוא עוטף בדוגמה שלהן:

- אני עוטף את שלושת הקומפוננטות Box בקומפוננט מיכל מסג Stack ומוסיף את המאפיין `divider` אליו אני מעביר את הקומפוננט `Divider` של API MUI

התוצאה בדף

- ניתן לראות שנוסף קו חוצץ בין כל אלמנט ואלמנט שבתוך הקומפוננט `Stack`



Grid

The Material Design responsive layout grid adapts to screen size and orientation, ensuring consistency across layouts. It uses CSS's Flexible Box module for high flexibility.

<https://mui.com/material-ui/api/grid/>



Grid Layout

- הkomponent Grid עם המאפיין container מהוות המוביל לקומponentות Grid עם המאפיין item על מנת לקבוע רווח בין הילדים נשתמש במאפיין spacing בkomponent האב Grid container משתמש במאפיין item על komponentות Grid שם הילדים של komponent Grid עם המאפיין container להוב נשתמש בארכיטקטורה זאת על מנת להפוך את המידע שבתוך komponent האב Grid לוספונסיבי. בדוגמה שלහן קבענו שבגודל מסך sx כל item יתפוס רווח שונה

```
58 const LayoutComp = () => {  
59   return (  
60     <Grid container spacing={2}>  
61       <Grid item xs={3}>  
62         <Box sx={{ p: 2, backgroundColor: "success.light" }}>one</Box>  
63       </Grid>  
64       <Grid item xs={6}>  
65         <Box sx={{ p: 2, backgroundColor: "primary.light" }}>two</Box>  
66       </Grid>  
67       <Grid item xs={3}>  
68         <Box sx={{ p: 2, backgroundColor: "error.light" }}>three</Box>  
69       </Grid>  
70     </Grid>  
71   );  
72};
```

one

two

three

! komponent Grid container מחלקת את המספר/
האלמנט שמכיל אותה ל – 12 חלקים שוים

Responsive Grid

- נוכל לקבוע אורך שונה לכל קומפוננט לצד Grid item בהתאם לגודל המסך.
- גורר זאת על ידי הוספת מאפיינים המיצגים גודלים מסויימים

```
75 const LayoutComp = () => {  
76   return (  
77     <Container sx={{ mt: 2 }}>  
78       <Grid container spacing={2}>  
79         <Grid item xs={12} sm={6} md={4} lg={3}> ◀  
80           <Box sx={{ p: 2, backgroundColor: "success.light" }}>one</Box>  
81         </Grid>  
82         <Grid item xs={12} sm={6} md={4} lg={3}>  
83           <Box sx={{ p: 2, backgroundColor: "primary.light" }}>two</Box>  
84         </Grid>  
85         <Grid item xs={12} sm={6} md={4} lg={3}>  
86           <Box sx={{ p: 2, backgroundColor: "error.light" }}>three</Box>  
87         </Grid>  
88         <Grid item xs={12} sm={6} md={4} lg={3}>  
89           <Box sx={{ p: 2, backgroundColor: "secondary.light" }}>four</Box>  
90         </Grid>  
91       </Grid>  
92     </Container>  
93   );  
94 };
```



! כל גודלי המסך השונים נמצאים בשקף הבא



Measurement Screen Units

No.	Unit	Detail	Screen Size
1.	xs	extra-small	0px-599px
2.	sm	small	600px-899px
3.	md	medium	900px-1199px
4.	lg	large	1200px-1536px
5.	xl	extra-large	1536px - above

```
97 const LayoutComp = () => {
98   return (
99     <Container sx={{ mt: 2 }}>
100       <Grid container rowSpacing={4} columnSpacing={1}>
101         <Grid item xs={12} sm={6} md={4} lg={3}>
102           <Box sx={{ p: 2, backgroundColor: "success.light" }}>one</Box>
103         </Grid>
104         <Grid item xs={12} sm={6} md={4} lg={3}>
105           <Box sx={{ p: 2, backgroundColor: "primary.light" }}>two</Box>
106         </Grid>
107         <Grid item xs={12} sm={6} md={4} lg={3}>
108           <Box sx={{ p: 2, backgroundColor: "error.light" }}>three</Box>
109         </Grid>
110         <Grid item xs={12} sm={6} md={4} lg={3}>
111           <Box sx={{ p: 2, backgroundColor: "secondary.light" }}>four</Box>
112         </Grid>
113       </Grid>
114     </Container>
115   );
116 }
```



Grid Spacing

אם נהייה מעוניינים ברווח שונה בין השורות
והעמודות בין התוכן נוכל להגדיר אותם
בנפרד

- הרוח בין השורות - rowSpacing
- הרוח בין העמודות - columnSpacing

Grid direction

```
121 <Container sx={{ mt: 2 }}>  
122   <Grid container spacing={1} direction="row-reverse"> ←  
123     <Grid item xs={12} sm={6} md={4} lg={3}>  
124       <Box sx={{ p: 2, backgroundColor: "success.light" }}>one</Box>  
125     </Grid>  
126     <Grid item xs={12} sm={6} md={4} lg={3}>  
127       <Box sx={{ p: 2, backgroundColor: "primary.light" }}>two</Box>  
128     </Grid>  
129     <Grid item xs={12} sm={6} md={4} lg={3}>  
130       <Box sx={{ p: 2, backgroundColor: "error.light" }}>three</Box>  
131     </Grid>  
132     <Grid item xs={12} sm={6} md={4} lg={3}>  
133       <Box sx={{ p: 2, backgroundColor: "secondary.light" }}>four</Box>  
134     </Grid>  
135   </Grid>  
136 </Container>  
137 );  
138 };
```



יש ביכולתנו להפוך את סדר הופעת האלמנטים באמצעות המאפיין direction שאותו נוסיף לקומponent Grid שעוטפת את התוכן

- row - הערך הדיפולטיבי משמאל לימין
- row-reverse – מימין לשמאל
- column - סדר עולה של האלמנטים
- column-reverse - סדר יורד של האלמנטים

התוצאה בדף:

- בגלל שקבענו סדר הפוך של התוכן ניתן לראות כי האלמנטים מסודרים מצד ימין לשמאלי

Justify-content

סידור התוכן במיכל

- center - מרכז התוכן
- flex-start – הצמדה לשמאלי (אם לא שינו את מאפיין direction ל - row)
- flex-end – הצמדה לימני (אם לא שינו את מאפיין direction ל - row-reverse)
- space-between – אלמנט ראשון מוצמד לשמאלי, אלמנט אחרון מוצמד לימין וכל שאר האלמנטים עם רווחים שווים ביניהם
- space-around – רווחים שווים בין כל האלמנטים
- space-evenly – רווחים שווים בין כל האלמנטים כולל מההתחלה וסוף המספר

! דוגמאות לכל סוג justify-content ניתן
לראות בקישור הבא

```
<Container
  sx={{ mt: 2, backgroundColor: "secondary.light", p: 2, height: 200 }}>
  <Grid container justifyContent="center"> ...
  </Grid>
</Container>
```



```
<Container
  sx={{ mt: 2, backgroundColor: "secondary.light", p: 2, height: 200 }}>
  <Grid container justifyContent="space-between"> ...
  </Grid>
</Container>
```



Align-items

סידור התוכן במילוי בצורה אנכית

- center - ימרכז את התוכן

- flex-start - יצמיד למעלה

- flex-end - יצמיד למטה

- Stretch - ימתוח את התוכן מלמעלה עד
למטה

- Baseline - יצמיד למאלה אבל יסדר אותם
באופן ממורכב

```
233 const LayoutComp = () => {
234   return (
235     <>
236       <Container sx={{ mt: 2, backgroundColor: "secondary.light", p: 2 }}>
237         <Grid
238           container
239           justifyContent="center"
240           alignItems="center" ←
241           sx={{ height: 200 }}>
242             <Grid item xs={12} sm={6} md={4} lg={3}>
243               <Box sx={{ p: 2, backgroundColor: "success.light" }}>one</Box>
244             </Grid>
245             <Grid item xs={12} sm={6} md={4} lg={3}>
246               <Box sx={{ p: 2, backgroundColor: "primary.light" }}>two</Box>
247             </Grid>
248             <Grid item xs={12} sm={6} md={4} lg={3}>
249               <Box sx={{ p: 2, backgroundColor: "error.light" }}>three</Box>
250             </Grid>
251           </Grid>
252         </Container>
253       </>
254     );
255   };
};
```





משימת Layout

Business-cards-app

- הפור את תצוגת הכרטיסים בקומפוננט Cards.js.jsx לrsponsive באמצעות הקומפוננט Grid של UI

Navigation

Components that are responsible for the design and logic of navigation menus





AppBar

"The top App bar provides content and actions related to the current screen. It's used for branding, screen titles, navigation, and actions"

[mui.com](https://mui.com/material-ui/api/app-bar/)

<https://mui.com/material-ui/api/app-bar/>



AppBar

קומponent זה משמש במייצל לתפריט נייר

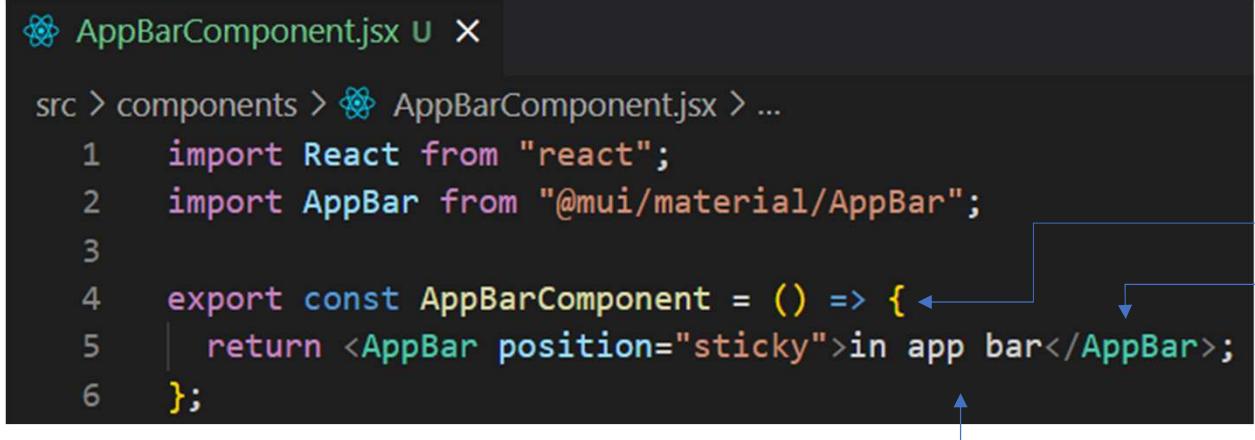
דוגמה שלהן:

- יצרנו קומponent בשם AppBarComponent

- הציבנו את הקומponent AppBar בתוכה
- קבענו שהתפריט יהיה position="sticky"
- ככלומר נראה אותו הוא מיקום בדף גם אם נגלה תוך מהו, אולם הוא לא יסתיר את תוכן הדף שמטהינו
- הציבנו מחרזת תווים בפתח children
- באובייקט props של הקומponent

- התוצאה בדף

! לרשימת הערךם למאפיין position ומאפיינים נוספים
<https://mui.com/material-ui/api/app-bar/#props>



```
AppBarComponent.jsx
src > components > AppBarComponent.jsx > ...
1  import React from "react";
2  import AppBar from "@mui/material/AppBar";
3
4  export const AppBarComponent = () => {
5    return <AppBar position="sticky">in app bar</AppBar>;
6  };

```

in app bar

ToolBar

קומפוננט זה ייתן הגדרות יצוב דיפולטיביות לתפריט הנביון בגין padding align-items ובודומה

בדוגמה שלהן:

- הציבנו את קומפוננט Toolbar בתוך AppBar
- העברנו מחרוזת תווים כchildren לkomponent Toolbar
- התוצאה בדף

```
93 import React from "react";
94 import AppBar from "@mui/material/AppBar";
95 import Toolbar from "@mui/material/Toolbar";
96
97 export const AppBarComponent = () => {
98   return (
99     <AppBar position="sticky">
100       <Toolbar>in Toolbar</Toolbar>←
101       </AppBar>↑
102     );
103   };

```

in Toolbar

! לרשימת מאפיינים נוספים

<https://mui.com/material-ui/api/toolbar/>



Bottom Navigation

"The Bottom Navigation bar allows movement between primary destinations in an app."

[mui.com](https://mui.com/material-ui/react-bottom-navigation/)

<https://mui.com/material-ui/react-bottom-navigation/>



Bottom Navigation

ואם המצא קומפוננטות מיוחדות במיוחד לעיצוב תפריט הניווט התחתון המיועד בעיקר ל视窗 טלפונים ניידים

- ניבא את הקומפוננטות והאייקונים החדשניים לנו מספרית ו-MUI

■ נקבע שהקומponent Paper של MUI תהווה המיכל לתפריט הניווט התחתון ונקבע כי הפוזיציה שלה תהיה sticky ב-

שיהיא תהיה במקומ קבוע בדף אך לא תעלה על הטקסט בדף

■ השתמש בקומponent BottomNavigation של MUI כמיכל של כפתורי הניווט

- כל Link/כפתור ניוט יהיה בתוך BottomNavigationAction הקומponent שמקבל את המאפיינים:

- label - הכתוב על כפתור הניווט
- icon – מקבל קומponent MUI

! **רשימת המאפיינים של BottomNavigationAction** <https://mui.com/material-ui/api/bottom-navigation-action/>

MuiBottomNavigators.jsx

```
client > src > sandbox > layout > footer > MuiBottomNavigators.jsx > ...  
1 import React from "react";  
2 import BottomNavigation from "@mui/material/BottomNavigation";  
3 import BottomNavigationAction from "@mui/material/BottomNavigationAction";  
4 import Paper from "@mui/material/Paper";  
5 import FacebookIcon from "@mui/icons-material/Facebook";  
6 import InstagramIcon from "@mui/icons-material/Instagram";  
7 import TwitterIcon from "@mui/icons-material/Twitter";  
8  
9 const MuiBottomNavigators = () => {  
10   return (  
11     <div style={{ marginTop: 8 }}>  
12       <div>  
13         Lorem ipsum dolor sit amet consectetur, adipisicing elit. Aperiam  
14         consequuntur est, recusandae facere odit ea repudiandae assumenda  
15         necessitatibus repellat debitis dolorem. Possimus qui animi impedit.  
16       </div>  
17  
18       <Paper sx={{ position: "sticky" }} elevation={4}> ←  
19         <BottomNavigation> ←  
20           <BottomNavigationAction label="Facebook" icon={<FacebookIcon />} />  
21           <BottomNavigationAction label="Instagram" icon={<InstagramIcon />} />  
22           <BottomNavigationAction label="Twitter" icon={<TwitterIcon />} />  
23         </BottomNavigation>  
24       </Paper>  
25     </div>  
26   );  
27 };  
28  
29 export default MuiBottomNavigators;
```

התוצאה בדף:

ניתן לראות כי תפריט הניווט התחתון מתחילה
במקום בו הטקסט נגמר בגל שקבענו על
הקומponent Paper שהוא יהיה
position="sticky"

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Aperiam consequuntur est,
recusandae facere odit ea repudiandae assumenda necessitatibus repellat debitis dolorem.
Possimus qui animi impedit.



```
29 const MuiBottomNavigators = () => {
30   return (
31     <div style={{ marginTop: 8 }}>
32       <div>
33         Lorem ipsum dolor sit amet consectetur, adipisicing elit. Aperiam
34         consequuntur est, recusandae facere odit ea repudiandae assumenda
35         necessitatibus repellat debitis dolorem. Possimus qui animi impedit.
36       </div>
37
38       <Paper sx={{ position: "sticky" }} elevation={4}>
39         <BottomNavigation showLabels>←
40           <BottomNavigationAction label="Facebook" icon={<FacebookIcon />} />
41           <BottomNavigationAction label="instagram" icon={<InstagramIcon />} />
42           <BottomNavigationAction label="twitter" icon={<TwitterIcon />} />
43         </BottomNavigation>
44       </Paper>
45     </div>
46   );
47 }
```

showLabels

MUI המזיא קומפוננטות מיוחדות במיוחד לעיצוב
תפריט הניווט התחתון המיעוד בעיקר
لتצוגת טלפונים ניידים

- אם נהיה מעוניינים לראות את שמות
כפתרונות הניווט נוסף את המאפיין
showLabels

! **רשימת המאפיינים של BottomNavigation**
<https://mui.com/material-ui/api/bottom-navigation/>

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Aperiam consequuntur est, recusandae facere odit ea repudiandae assumenda necessitatibus repellat debitis dolorem. Possimus qui animi impedit.



Facebook



instagram



twitter

התוצאה בדף:

ניתן לראות ששמותיהם בתוך המאפיין
מופיעים מתחתי לתמונה של האיקונים

משימת navigation

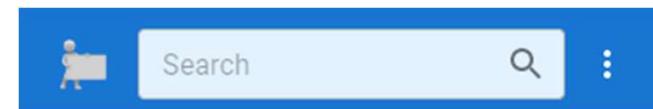


Business-cards-app

- צור תפריט עליון לאפליקציה שבגודל מסך md ומעלה יראה כה:

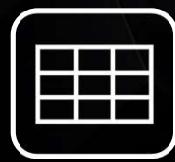


- ובתצוגת טלפון סלולרי יראה כה:



- צור תפריט ניוט תחתון שיראה כה:





Menu

Too...

[/https://mui.com/material-ui/customization/typography](https://mui.com/material-ui/customization/typography)



Menu

ישנים ערכיהם זהה.

- הוספנו ל-

! לרשימת ה-ערכיהם דיפולטיביים לטקסט בリンק הבא
<https://mui.com/material-ui/api/list/>

משימת Menu



- כוורת מטנה

Form & Inputs

Form & Inputs ...

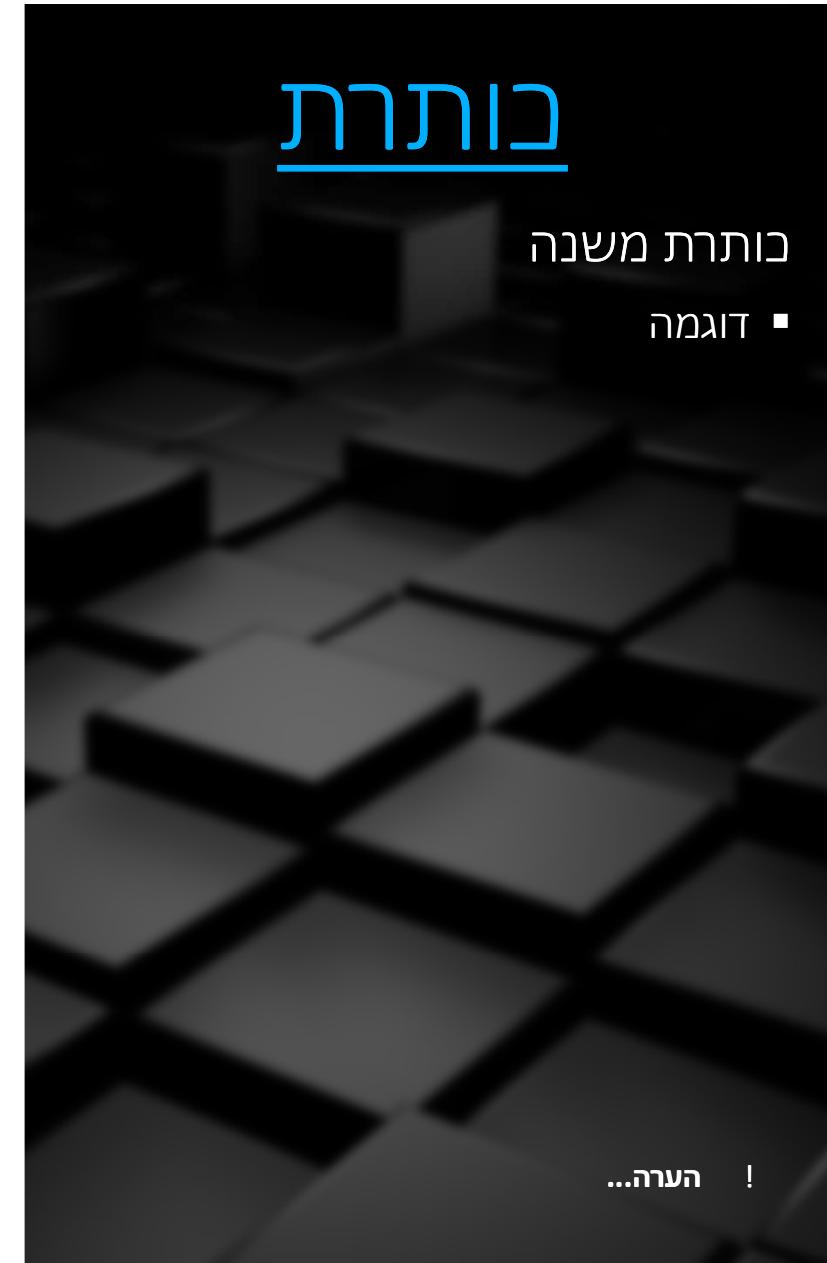


כותרת

כותרת משנה

■ דוגמה

הערה... !



משימת Media



- כוורת מטנה

MUI THEMES

Form & Inputs ...

[/https://mui.com/material-ui/customization/theming](https://mui.com/material-ui/customization/theming)



משימת Bottom Navigation



- כוורת מושנה