**HTML5**

**מעבדה**

**פריסות – flex ו- grid**

# תרגיל 1

נתון קובץ cats2.html המכיל את כל התכנים הנחוצים. **אין לשנות את הקובץ** (חוץ מתמונות ופונטים). **חובה לעבוד לפי ההנחיות המופיעות אחרי תיאור הדף**.

במידה ואין לך אינטרנט, תוכל להחליף את קישורי התמונות לתמונות מתוך התיקייה images.

תיאור הדף

1. הדף הסופי צריך להיראות באופן הבא:



1. במצב של hover על הלינק בסרגל הניווט, הלינק יקבל opacity של 0.7
2. במצב של hover על תמונה, התמונה תראה באופן הבא (שינוי צבע והופעת הטקסט):



1. במסך קטן מ 600px הדף יראה באופן הבא (בתמונה רואים בצד ימין את החלק העליון של הדף ובצד שמאל את החלק התחתון של הדף – תמונות אנכיות):



ההנחיות (אל תדלג על אף סעיף, בדוק בכל שלב מה אתה מקבל ותבין למה זה קורה).

הבניה תתבצע בשלבים הבאים: הגדרות כלליות, הגדרת טיפוגרפיה, הגדרת מבנה של שלד הדף, כניסה לאלמנטים פרטניים, הגדרה עבור מסכים קטנים.

לפני שתתחיל בדוק איך הדף נראה ללא העיצוב!

תהנה מהעשייה 😊

1. הגדרות כלליות
2. הוסף תיקיית עיצוב לאזור העבודה. העיצוב חייב להיות **חיצוני** **בתיקייה נפרדת** לקובץ cats2.html (בדוק את שם התיקייה כפי שמוגדר בקישור בתוך קובץ html). הקובץ html כבר כולל קישור ל- normalize.css, אשר חייב להיות מקושר לפני קישור לקבצי css שלנו. מטרת הקובץ ליישר קו בין כל הדפדפנים כך שהתצוגה הראשונית תהיה זהה בכולם. למידע נוסף בדוק ב- <https://necolas.github.io/normalize.css/>  
   בנוסף ישנו קישור לאחד הפונטים של google. אתה רשאי להשתמש בפונטים אחרים – מותר להוסיף קישור לפונט שתבחר באזור ה- head של html או תשתמש בכל פונט אחר שתבחר תוך ציון משפחה כללית (כמו serif, sans-serif...).
3. בשלב זה נוסיף לקובץ css הגדרות כלליות:

הגדר בתחילת העיצוב את ה- \*– חובה! חובה בכל התרגילים בכל המעבדות!

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* General

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

\*, \*::after, \*::before {

box-sizing: border-box;

margin: 0;

padding: 0;

}

1. הגדר לכל הדף גודל של פונט בסיס 10px (62.5%):

:root {

font-size: 62.5%;

}

1. עכשיו הגדר טיפוגרפיה: הגדר מחסנית של פונטים (font-family), גדלים לכל האלמנטים (font-size), משקל הפונט (font-weight), מרחק בין שורות (line-height), מרחק בין אותיות (letter-spacing), צבעים ... **הגדר את גודל הפונט לאלמנטים ביחידות של em או rem (רצוי rem)**. אין להשתמש בפיקסלים עבור גודל הפונט.

בדוק מה קיבלת והמשך אם אתה מרוצה מהתוצאה. תוכל לחזור לכאן בהמשך לתיקונים נוספים.

1. הגדרת מבנה של הדף: בשלב זה נמקם את החלקים העיקריים על הדף. ניתן לראות שהדף מחולק לשלושה חלקים. header, main, footer:



אנחנו נרצה שה- footer תמיד יהיה למטה (גם אם החלק המרכזי הוא קצר מאוד). לצורך כך נשתמש ב- flex כפי שלמדנו עבור ה- body.

הוסף מסגרות לכל אחד משלושת החלקים והגדר באופן הבא (הורד את כל התוכן של ה- main כדי שתוכל לראות את החלוקה):

/\* major blocks: header, main/footer positioning \*/

body {

min-height: 100vh;

display: flex;

flex-direction: column;

}

header {

border: 2px solid red;

}

footer {

border: 2px solid blue;

}

main {

border: 2px solid green;

flex-grow: 1;

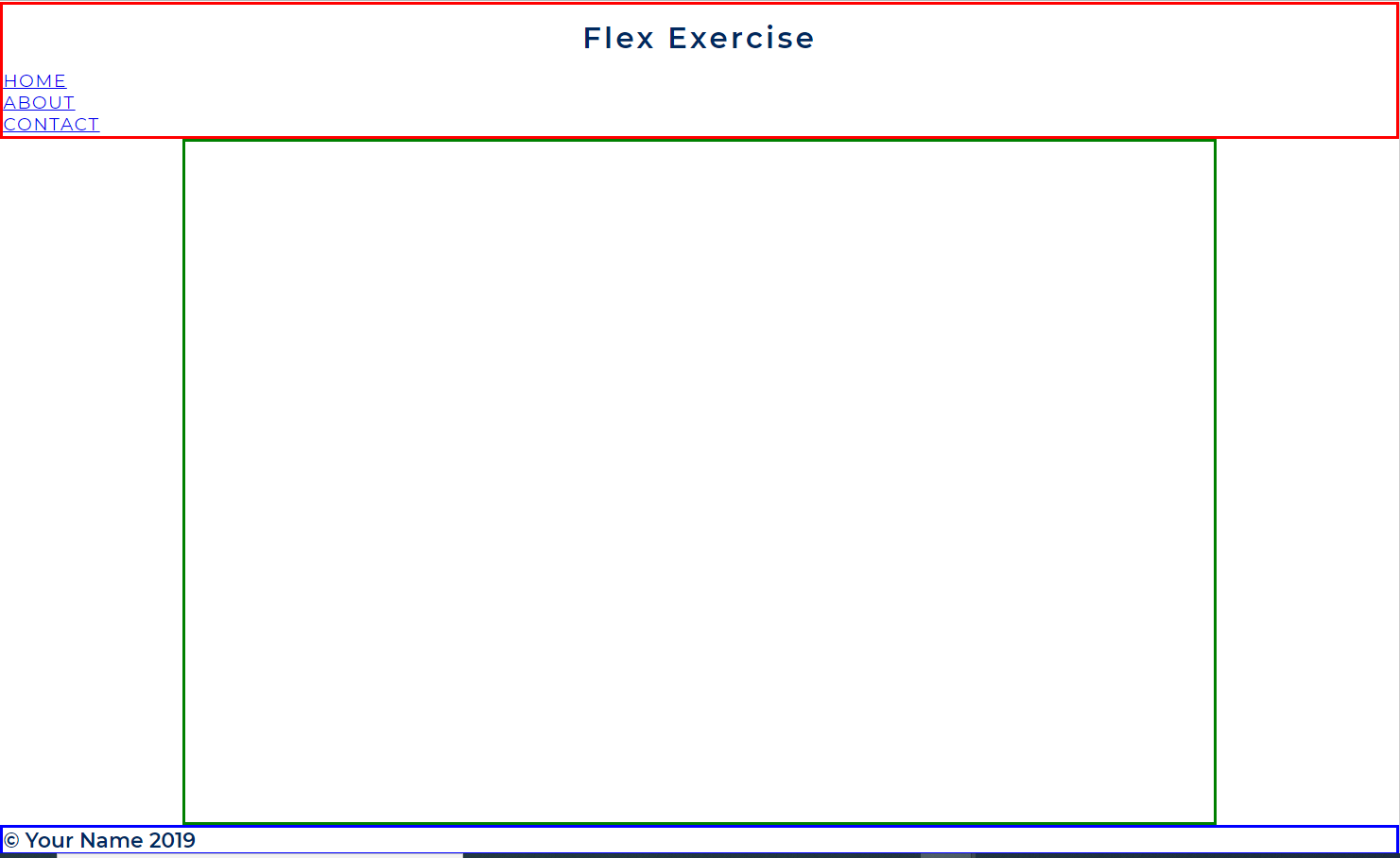
width: 80%;

max-width: 700px;

margin: 0 auto;

}

שים לב לשימוש ב- max-width (כדי שהחלק המרכזי לא יהיה רחב מדי במסכים גדולים), flex-grow למריחה של ה- main עד לאזור של ה- footer, margin שמאל וימין auto למרכוז ה- main ושימוש באחוזים עבור רספונסיביות. ה- body חייב להכיל את min-height: 100vh כדי שה- footer יתמקם בתחתית הדף במידה והחלק המרכזי קצר. מה שקיבלנו זה מילוי של כל המסך באופן הבא:



1. סדר את ה- footer ואת ה- header. ב- header השתמש ב- flex ל- nav כדי למרכז את הקישורים.

/\* header \*/

nav ul {

display: flex;

justify-content: center;

}

nav li a {

padding: .8em .4em;

margin: 0 10px;

}

השלם את עיצוב הקישורים ומצב של hover על הקישורים. תקן את הטיפוגרפיה במידת הצורך.

1. עכשיו עוברים לסידור התמונות. החזר את תוכן ה- main ל- html.

נשתמש בהגדרה flex ל- main כדי למקם את התמונות אופקית ממורכזות:

display: flex;

align-items: center; /\* vertical centering \*/

justify-content: space-between;

ניתן רוחב של 31% לכל box וגובה נניח 300px וקצת margin בין הקופסאות.

את התמונה עצמה בתוך ה- box נמקם באופן הבא:

img {

display: block; /\* MUST!!!! \*/

width: 100%;

height: 100%;

object-fit: cover; /\* fill the box \*/

object-position: center top; /\* not must – try to remove \*/

}

חובה להגדיר את התמונה כ- blockאו inline-block.

את ה- text הגדר במיקום אבסולוטי וגם כ- flex:

.text {

position: absolute;

top: 0;

left: 0;

width: 100%;

height: 100%;

padding: 2em;

display: flex;

justify-content: flex-end;

flex-direction: column;

}

השלם את כל ההגדרות כולל hover כך שהטקסט יופיע רק על hover. הוסף box-shadow במצב של hover. את צבע הרקע במצב hover אפשר להוסיף ע"י הוספת שכבה מעל לקופסה ע"י שימוש בבורר ::after, מיקומו אבסולוטי על כל הקופסה וביצוע blending (שימוש ב- mix-blend-mode). לא חובה עבור תרגיל זה.

השלם טיפוגרפיה של הטקסט במידת הצורך.

עבודה עם מסכים קטנים:

אחרי שסיימת והדף נראה טוב, בדוק איך הוא יראה ב- Galaxy S5 (השתמש ב- F12 ובחר את ה- Galaxy).

עכשיו הורד בקובץ HTML את השורה:

<meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">

ובדוק שוב ב- Galaxy. השורה הזאת הכרחית לעבודה עם ה – mobile.

עכשיו נסדר קצת ל- mobile. למשל נקטין את גודל הפונט למסכים שרוחבם קטן או שווה ל- 600px של CSS). בחלק תחתון של העיצוב נכתוב את ההגדרות הבאות:

*@media* only screen and (max-width: 600*px*) {

:root {

font-size: 50%;

}

}

בדוק שגודל הפונט ב- Galaxy עכשיו קטן יותר (בהנחה שהגדרת את כל הגדלים ב- em או rem). אתה יכול לשנות גם תכונות אחרות באלמנטים שונים.

זה נקרא media query. בדוק באינטרנט מה זה.

נסה להוריד את רוחב של main תוך כדי זה שאתה משאיר את רוחב המקסימלי. בדוק מה אתה מקבל ב- Galaxy. החזר – ובדוק שוב. לעיתים נגדיר רק את הרוחב המקסימלי וזה יעבוד טוב בכל סוגי המסכים (גם גדולים וגם קטנים). במסכים שהם קטנים מגודל המקסימלי, מיכל יתפוס את כל הרוחב. נוכל להוסיף קצת margin או padding כדי להרחיק את התוכן מהצדדים.

לצורך התרגיל, שנה את מיקום התמונות מאופקית לאנכית:

@media (max-width: 600px) {

main {

flex-direction: column; /\* vertical direction of images \*/

}

.box {

height: 200px;

width: 80%;

margin: 5px;

}

}

**עבודה נעימה!**