**HTML5**

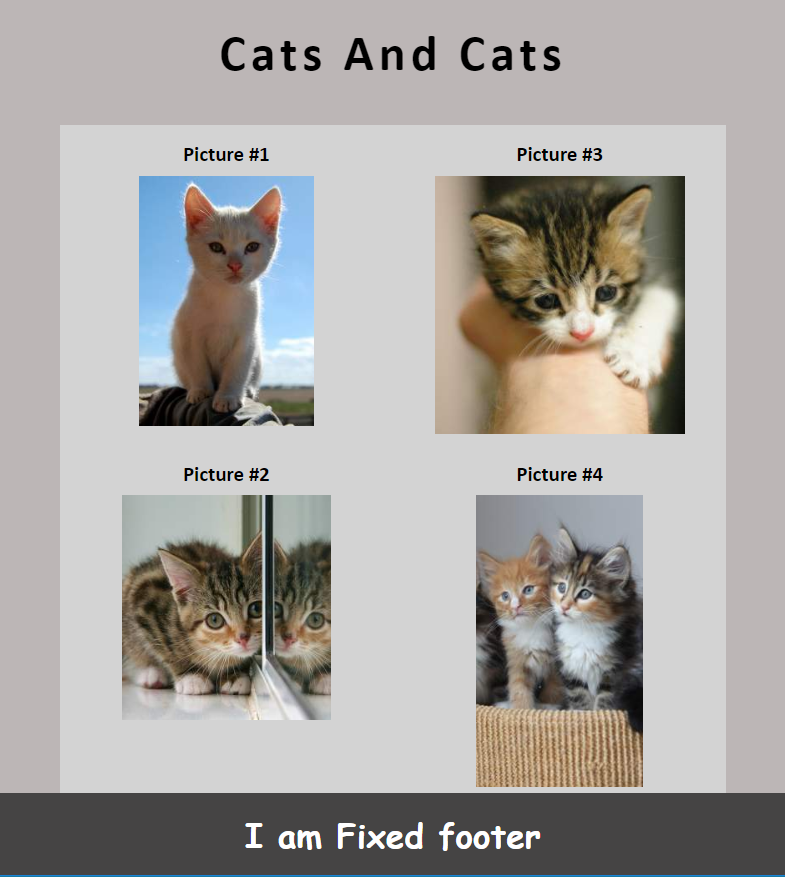
**מעבדה**

**פריסות – flex/grid**

# במעבדה זו 2 תרגילים. לפני שתתחיל, בדוק איך הם נראים ללא העיצוב!

# תרגיל 1

הוסף עיצוב **חיצוני** **בתיקייה נפרדת** לקובץ cats1.html שיראה כמו בתמונה למטה (בתמונה רואים את החלק העליון של scroll, ישנן סה"כ 6 תמונות, 2 תמונות בשורה). שים לב להנחיות הבאות (בדוק בכל שלב מה אתה מקבל ותבין למה זה קורה).



1. **אין לשנות את קובץ ה- html** מלבד הכנסת קובץ CSS! במידה ואין לך אינטרנט, תוכל להחליף את קישורי התמונות לתמונות בתוך התיקייה images.
2. הגדר בתחילת העיצוב את ה- \* כמו במעבדה הקודמת – חובה! חובה בכל התרגילים בכל המעבדות!

\* {

box-sizing: border-box;

margin: 0;

padding: 0;

}

1. הגדר ל- body גודל של פונט (כמו שלמדנו עם שימוש ב- :root) ואת כל גדלי הפונט לאלמנטים הגדר ביחידות של rem
2. בחלק התחתון ישנו fixed footer שלא זז עם scrolling.

position: fixed;

bottom: 0;

left: 0;

width: 100*%*;

הגדר אותו כ- flex ומרכז אופקית ואנכית את מה שיש בתוכו

display: flex;

align-items: center;

justify-content: center;

הגדר ל- footer תכונות נוספות כגודל טקסט, גובה (תן כ- 100px) וכו'

1. ל- container תן רוחב של 80%, max-width 800px ומרכז אותו באמצע הדף אופקית (השתמש ב- margin). הגדר לו margin-top כ- 50px ו- margin-bottom כ- 100px (כגובה של footer – תחשוב למה).
2. מרכז את כל הכותרות (גם של תמונות) ע"י text-align. את מרכוז הכותרות של תמונות הגדר בתוך ה-

בורר של picture-group.

1. הפוך את הכותרת העליונה לאותיות גדולות (text-transform: uppercase) או אחר לפי רצונך.
2. השתמש ב- flexbox ל- container לסדור של תמונות בתוך ה- container. שים לב שיש שתי תמונות בשורה. דאג ל- wrap

*.container* {

display: flex;

flex-wrap: wrap;

1. הגדר רוחב של picture-group כ- 50% לפחות. כך יכנסו ל- container ברוחב 2 תמונות בדיוק אם יש מספיק מקום. עשה זאת בצורה הבאה:

*.picture-group* {

flex: 1 1 50*%*;

1. הגדר רקע ל- container
2. הגדר רקע אפור לכל הדף.
3. בנה דף נוסף הזהה למה שעשית בתרגיל זה. רק שהפעם אל תגדיר את ה- footer כ- fixed, אלא השתמש ב- flex-grow של main כפי שלמדנו בכיתה

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* (max-width: 600*px*) {

:root {

font-size: 50%;

}

}

בדוק שגודל הפונט ב- Galaxy עכשיו קטן יותר. אתה יכול לשנות גם תכונות אחרות באלמנטים שונים.

זה נקרא media query. בדוק באינטרנט מה זה.

נסה למחוק את רוחב של container 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.

# תרגיל 2

הוסף עיצוב **חיצוני** **בתיקייה נפרדת** לקובץ coffee.html שיראה כמו בתמונה הבאה.

שים לב להנחיות הבאות:

1. **אין לשנות את קובץ ה- html**
2. הגדר \* וגודל הפונט ב- body כמו בתרגיל הקודם. זכור לעשות זאת תמיד בכל עבודה גם אם לא אציין זאת.
3. ל- container תן רוחב של 80% (באופן זה הדף יהיה responsive ) ומרכז אותו באמצע הדף אופקית (זוכר margin?). הגדר לו margin-top כ- 50px. רצוי גם להגדיר רוחב מקסימלי (max-width).
4. הגדר רקע בהיר לכל הדף (ל- body). הגדר רקעים מתאימים בתוך המיכל
5. במקרה זה נוח להסתכל על המיכל כ- grid עם 3 עמודות. כל עמודה ניתן להגדיר כ- flex עם כיוון של עמודות. הגדר את העמודות במיכל כ-:

display: grid;

grid-template-columns: 100*px* auto 20*%*;

שים לב שפריסה זו לא מתאימה למוביל. במקרה של מוביל יש לעצב אחרת תוך כדי שימוש ב- media query (כמו שעשינו בתרגיל הקודם). בתרגיל זה לא נבצע זאת כי דורש שינוי רציני של סידור האלמנטים.



**עבודה נעימה!**