הטכנולוגיה העכשווית של הפרדת צבעים, מאפשרת הצגה דיגיטלית של תמונות, באופן חד וחלק, הרבה יותר מאשר בעבר. אחד הגורמים המרכזיים המאפשרים עובדה זו, הוא יכולת הפרדת הצבעים של מסכי התצוגה של היום. יכולת זו גבוהה פי כמה מונים מיכולת הפרדת הצבעים במסכים מדורות קודמים. ישנם גורמים נוספים המעורבים בחדות התצוגה (כגון: רכיבי זכרון משופרים, יכולות עבוד תמונה גבוהות ועוד גרומים רבים נוספים), אולם ללא אחידות בהפרדת הצבעים הבסיסית, יכולות אלה לא יבואו לכדי מימוש אופטימלי. הדבר נכון הן למסכים בקולנוע, הן למסכי טלויזיה והן למסכי מחשב. כאן, לא נעסוק במסכי קולנוע ולא בשיטות שידור שונות כמו PAL, NTSC, FullHD וכד׳ (למרות שנושא זה קשור בקשר הדוק ליכולת ושיטת הפרדת הצבעים, ולמרות שפיתוחים בתחום הפרדת הצבעים ופיתוחים בתחום שיטות השידור תורמים הדדית אלה לאלה) אלא ביכולת ההפרדה הבסיסית המאפשרת מגוון מסוים של צבעים וגוונים המובחנים זה מזה.

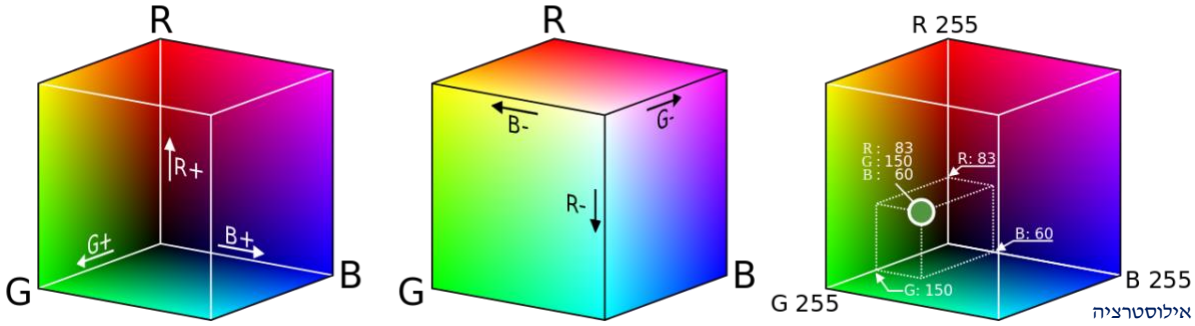
מסכים עכשוויים מסוגלים להבחין, לכל הפחות, בין עשרות מיליוני צבעים וגוונים שונים (המספר המינימלי המדויק הוא $256^3 = 16,777,216$). תצוגה של כל הצבעים והגוונים הללו יחד, אפילו אם מדובר בגרצג (פיקסל בלעז) אחד עבור כל גוון, היא מסובכת, גוזלת זמן ומשאבי עיבוד וזכרון רבים, ולמעשה היא מיותרת כשלעצמה. המשמעות המעשית של מספר רב של גוונים וצבעים, היא בהבחנה ביניהם ובמעברים ביניהם. זה מה שמאפשר בעיקר תמונה חדה, ברורה וחלקה. אם נצליח להציג יחד את כל מיליוני הגוונים, נוכל אמנם לראות מעבר חלק בין הגוונים, אולם לעין האנושית יהיה קשה מאוד להבדיל בין גוונים מסוימים הקרובים מאוד זה לזה. אי-לכך, אין בכוונתי לנסות להציג מייצג כזה, אלא להגיע להבנה בסיסית ביותר של שיטות הפרדת הצבע השונות וההבדלים העיקריים ביניהן.

RGB – ההפרדה הבסיסית (הן היסטורית והן תפיסתית) של צבעים היא לפי הכמות היחסית של אדום, ירוק וכחול (Red, Green, Blue) בכל צבע וגוון. הפרדה זו מתאימה, הן להבחנה בין אורכי הגל השונים של כל צבע וצבע והן לאופן בו העין והמוח האנושיים מבחינים בצבע (הקולטנים השונים של צבע, בין עדשת העין והחלק במוח המפענח צבע, קולטים את אורכי הגל הנפרדים, של אדום, ירוק וכחול). כולנו למדנו, כבר בגן הילדים, שצבעי היסוד הם אדום, **צהוב** וכחול ולא אדום, ירוק וכחול. להבחנה זו נתייחס בהמשך, אולם מבחינה טכנולוגית, הן של העין והמוח האנושיים והן של רכיבים אלקטרוניים, ההפרדה של אדום, ירוק וכחול היא ההפרדה הפשוטה ביותר ליישום. כל גרצג בצג המחשב או הטלויזיה, מסוגל להקרין, עבור תמונה מסוימת, מידת הארה בגוון אדום, בגוון ירוק ובגוון כחול, בכדי ליצור את הצבע המתאים. שיטת ההפרדה הזו היא שיטה אדפטיבית של הפרדת צבע. אנו מוסיפים הארה של קרן אור בכל אחד משלושת הגוונים, זו לזו, בכדי ליצור את הצבע המבוקש, כך שאם נוסיף זה לזה, את המידות המקסימליות של הארה, בכל אחת משלושת הקרניים. נקבל צבע לבן.

ניתן לכל אחת מהקרניים ערך מספרי בין 0 ל 255 (הערך המקסימלי שניתן לבטא בשמונה ספרות בינאריות. כלומר: בעזרת הספרות 0 ו 1 בשמונה מקומות שונים, ניתן

מסמך זה נכתב ,ללא שום אחריות מצד הכותב! האחריות על השימוש בתוכן זה חלה כולה על המשתמש בו! ע״י רונן גיל . הרישיון לשימוש במסמך זה כפוף לתנאי הרישיון המופיעים מטה. חלק מהזכויות שמורות.
מסמך זה מופץ תחת רישיון ייחוס, שימוש לא מסחרי, איסור יצירות נגזרות 2.5 ישראל

להגיע ל 256 צירופים נבדלים זה מזה) ונסדר את הערכים בקובייה כך שכל ציר מבטא את כמות ההארה של קרן (אדום, ירוק, כחול) וכך כל חלק בקובייה יכול לקבל צבע מובחן אחד מתוך 256^3 = 16,777,216 צבעים שונים.



אילוסטרציה
של קובייית הצבעים בהפרדת RGB 24bit לקוחה מאתר מיזם ויקימדיה.

באמצעות מערכת קובייתית כזו, ניתן להבחין בין מספר רב מאוד של גוונים (אם נחלק כל אחד מהצירים ליותר מ 256 ערכים) או למספר מצומצם יותר של גוונים – כפי שעשו מערכות מחשב בעבר (העמוד retroColors.php מדגים כמה מההפרדות שהיו נהוגות במערכות מחשב בעבר, חלקן מבוססות RGB).

Hex – ביטוי קומפקטי יותר להפרדת הצבעים RGB ניתן לקבל, כאשר ממירים את הערכים של שלושת הצבעים למספרים בבסיס 16. מכיוון ש 16 הוא חזקה של 2 (2^4) הוא מתאים בדיוק לתיאור מספרים בינאריים ומכיוון שהמעריך בחזקה זו הוא 4 (מחצית מספר התווים בכל מספר בינארי בעל שמונה תווים), הרי שהוא מתאים באופן מושלם לתיאור הערכים ב RGB. אם נשתמש באותיות A, B, C, D, E, F במקום הספרות החסרות בין 10 ל 15, נוכל לתרגם כל ערך של כל אחד מהצבעים בעזרת שני תווים בלבד ובשישה תווים נוכל לבטא כל ערך במתחם RGB 24bit. אנו נוהגים להקדים את הקוד ההקסדצימלי בתו '#' בכדי לסמן כי מדובר בקוד RGB ולא במחרוזת תווים אקראית או במספר בבסיס עשרוני.

פונקציה פשוטה בפייתון יכולה להמיר ערכי RGB לקוד ההקסדצימלי התואם:

```
def RGB2HEX(R=0, G=0, B=0):
  return('#' + hex(R)[2:].upper() + hex(G)[2:].upper() + hex(B)[2:].upper())
```

CMYK – כאשר התמונה עוברת מצג המחשב אל המדפסת (או לכל אמצעי דפוס אחר), אין כל משמעות לעוצמת ההארה. לכן המודל האדפטיבי של RGB לא מתאים למצב זה. בדפוס אנו משתמשים במודל הפרדת צבעים מאוד דומה, אך במקום מודל אדפטיבי, זהו מודל רדוקטיבי. במקום להוסיף לעוצמת ההארה אנו מפחיתים מכמות הדיו בצבע מסוים. כאן בא לידי ביטוי צבע היסוד השלישי, הוא הצבע הצהוב. הפרדה זו, בין הצבעים ציאן (בין תכלת לטורקיז), מג'נטה (סוג של סגול בהיר) וירוק, אופטימלית לדפוס היות ודיו המורכב מכמויות יחסיות של צבעים אלה, יכול להרכיב כמעט כל צבע אפשרי, ועוד יותר היא מתאימה להמרה מתמונה דיגיטלית המוקרנת על הצג, היות וצבעים אלה נמצאים בדיוק באמצע, בין הצבעים אדום, ירוק וכחול. בהפרדה בדפוס, אנו מוסיפים גם את השחור (blacK) כצבע מפתח (Key) משלוש סיבות עיקריות:

מסמך זה נכתב ,ללא שום אחריות מצד הכותב! האחריות על השימוש בתוכן זה חלה כולה על המשתמש בו! ע"י רונן גיל . הרישיון לשימוש במסמך זה כפוף לתנאי הרישיון המופיעים מטה. חלק מהזכויות שמורות.
מסמך זה מופץ תחת רישיון ייחוס, שימוש לא מסחרי, איסור יצירות נגזרות 2.5 ישראל

1. כמודל הפרדה רדוקטיבי, אם נחבר את שלושת צבעי היסוד במודל CMYK יחד – נקבל צבע שחור.
2. בכדי לגוון עוד יותר את הצבעים המופקים מהדיו, ניתן להוסיף גוון אפור במידות שונות לכל גוון ולקבל מספר רב יותר של גוונים.
3. דיו שחור זול יותר לשימוש בדפוס, מדיו צבעוני. הוספת המפתח השחור לקוד הצבע, מאפשרת לחסוך בשימוש בדיו צבעוני ולהגביר במקומו את השימוש בדיו שחור.

מדפסות ומכונות דפוס אחרות, שונות זו מזו, כמו גם המדיה (נייר בד"כ) עליה מדפיסים. לכן, היישום של מודל ההפרדה CMYK שונה גם הוא בין מכשירים שונים ומדיות שונות, אולם ככלל, מודל זה עדיף לצורך הפרדת צבעים לדפוס, על-פני מודלים אחרים.

ציאן, מג'נטה וצהוב בלבד (משמאל).
התמונות מאתר מיזם ויקימדיה.

מסמך זה נכתב ,ללא שום אחריות מצד הכותב! האחריות על השימוש בתוכן זה חלה כולה על המשתמש בו! ע"י רונן גיל . הרישיון לשימוש במסמך זה כפוף לתנאי הרישיון המופיעים מטה. חלק מהזכויות שמורות.
מסמך זה מופץ תחת רישיון ייחוס, שימוש לא מסחרי, איסור יצירות נגזרות 2.5 ישראל

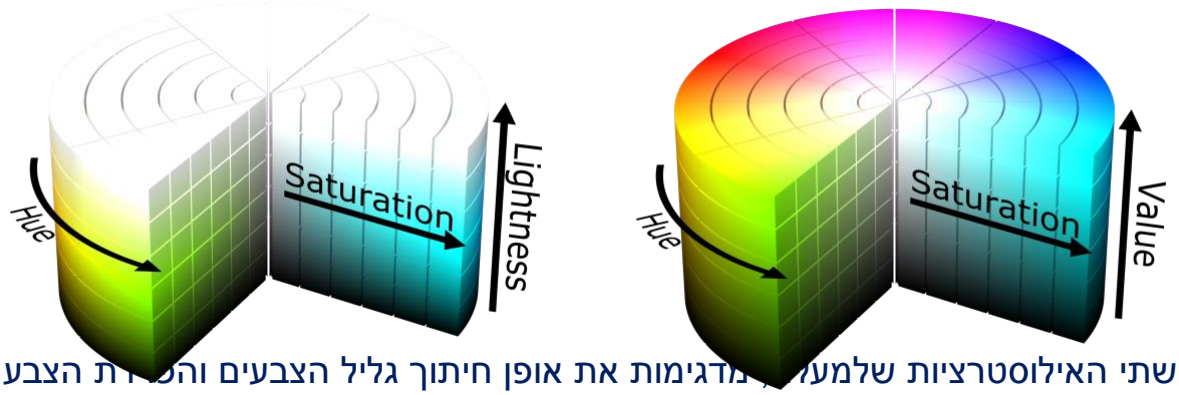
הקוד הבא הוא פונקציה בפייתון שנועדה להמיר ערכי RGB 24bit לאחוזי טיפת דיו ב CMYK:

```
def RGB2CMYK(R=0, G=0, B=0):
    r = R / 255.0 #
    g = G / 255.0 # Reducing rgb values from 8bit to Decimal percentile.
    b = B / 255.0 #

    if R == G and R == B and R == 0:
        return('(0%, 0%, 0%, 100%)') # Pure Black
    else:
        Max = max(r, g, b) # Evaluating the Maximum value.

        K = round((1 - Max), 4) # Calculating the Key(Black) value.
        if K < 0:
            K *= (-1)
        if K < 0.0050:
            K = 0.0000

        C = round(((1 - r - K) / (1 - K)), 4) # Calculating Cyan.
        if C < 0:
            C *= (-1)
        if C < 0.0050:
            C = 0.0000
        C *= 100

        M = round(((1 - g - K) / (1 - K)), 4) # Calculating Magenta.
        if M < 0:
            M *= (-1)
        if M < 0.0050:
            M = 0.0000
        M *= 100

        Y = round(((1 - b - K) / (1 - K)), 4) # Calculating Yellow.
        if Y < 0:
            Y *= (-1)
        if Y < 0.0050:
            Y = 0.0000
        Y *= 100

        K *= 100

        return('(' + str(C) + '%, ' + str(M) + '%, ' + str(Y) + '%, ' + str(K) + '%)')
```

מסמך זה נכתב ,ללא שום אחריות מצד הכותב! האחריות על השימוש בתוכן זה חלה כולה על המשתמש בו! ע"י רונן גיל . הרישיון לשימוש במסמך זה כפוף לתנאי הרישיון המופיעים מטה. חלק מהזכויות שמורות.
מסמך זה מופץ תחת רישיון ייחוס, שימוש לא מסחרי, איסור יצירות נגזרות 2.5 ישראל

HSL ו HSV – הפרדות הצבעים בהן עסקנו עד עתה, מבוססות על מודל של קובייה. מתוך הניסיון של אנשים העוסקים ביצירת גווני צבע, ללא קשר לטכנולוגיה אלקטרונית (בעיקרם ציירים וצַבָּעִים) פותחו שני מודלים להפרדת צבע, המבוססים על גליל במקום קובייה. בהפרדת הצבע לפי מודל HSL ו HSV , הצבעים השונים מסודרים במעגל חתך הבסיס של הגליל, כך שכל צבע נמצא בגזרה מסוימת של המעגל. ב 0^o (אפס מעלות) נמצא האדום שהופך עד 30^o לכתום. ב 60^o נמצא את הצהוב וב 120^o הירוק. ב 180^o נמצא הציאן וב 240^o הכחול. ב 300^o נמצא את המג׳נטה (שבמערכת זו נקראת בד״כ פוקסיה) וב 360^o נסגור את המעגל בחזרה עם האדום. גזרה זו (במעלות בטווח של 360 – 0), מוצבת בתור ערך הגוון (Hue) ואליו נוסיף עוד שני ערכים. בשני סגנונות ההפרדה הללו הערך הבא הוא רוויה (Saturation), אולם ההגדרה של רוויה זו, מעט שונה בין שתי שיטות ההפרדה. ב HSL הרוויה מחושבת לפי כמות האפור בצבע המבוקש, לעומת הצבע הנקי מאפור. ככל שיש יותר אפור בצבע, ערך הרוויה יקטן.

ב HSV אנו מחשבים את המעבר מהצבע המלא ללבן. ככל שנתקרב ללבן, כך ערך הרוויה יקטן. בשתי השיטות , עבור לבן, שחור וכל גוון של אפור, אין משמעות לערך הגוון (Hue) ולערך הרוויה (Saturation). לכן, בשתי השיטות ערכים אלו יהיו תמיד ׳0׳ בכל הגוונים שבין לבן לשחור. כמו כן, עבור כל הגוונים שבין אפור ללבן, כל הערכים יהיו זהים בשתי השיטות. הערך השלישי, הארה (Luminance) ב HSL הוא הציר שבין שחור, לצבע המלא וללבן. הצבע המלא מיוצג כ 50%. כל מה שמתחת ל 50% הוא פחות מואר (מתקרב לשחור) וכל מה שמעל 50% הוא יותר מואר (מתקרב יותר ללבן). ב HSV לעומת זאת, הערך השלישי, ששמו הוא פשוט ערך (Value), הוא הציר שבין הצבע הלא לשחור. 100% פירושו הצבע ללש גוון שחור כלל ו 0% פירושו שחור מלא. האילוסטרציה הבאה מדגימה את מדידת הערכים בחתך הגליל, ב HSL וב HSV.



שתי האילוסטרציות שלמעלה מדגימות את אופן חיתוך גליל הצבעים והכמות הצבע, לפי HSL (מצד שמאל) ו HSV (מצד ימין).
קבצי האילוסטרציה לקוחים מאתר מיזם ויקימדיה.

מסמך זה נכתב ,ללא שום אחריות מצד הכותב! האחריות על השימוש בתוכן זה חלה כולה על המשתמש בו! ע״י רונן גיל . הרישיון לשימוש במסמך זה כפוף לתנאי הרישיון המופיעים מטה. חלק מהזכויות שמורות.
מסמך זה מופץ תחת רישיון ייחוס, שימוש לא מסחרי, איסור יצירות נגזרות 2.5 ישראל

פונקציית פייתון להמרת ערכי RGB לערכי HSL*:

```
def RGB2HSL(R=0, G=0, B=0):
    r = R / 255.0 #
    g = G / 255.0 # Reducing rgb values from 8bit to Decimal percentile.
    b = B / 255.0 #

### Evaluating the Minimum and Maximum Values
    Min = min(r, g, b)
    Max = max(r, g, b)

### Calculating Luminance as a percentage.
    rawl = (Min + Max) / 2.0
    l = round((rawl * 100), 2)
    L = str(l) + '%'

    if Min == Max: # White or Black or a shade of pure Gray.
        S = str(0) + '%'
        H = str(0)
    else: # Calculating Saturation and Hue for all other Colors.
        if rawl < 0.5:
            raws = (Max-Min)/(Max+Min)
        else:
            raws = (Max-Min)/(2.0-Max+Min)
        s = round((raws * 100), 2)
        S = str(s) + '%'
        if Max == r:
            rawh = (g-b)/(Max-Min)
        elif Max == g:
            rawh = (b-r)/(Max-Min) + 2
        elif Max == b:
            rawh = (r-g)/(Max-Min) + 4
        h = int(round(rawh * 60))
        if h < 0:
            h += 360
        H = str(h)
        return('hsl(' + H + ', ' + S + ', ' + L + ')')
```

מסמך זה נכתב ,ללא שום אחריות מצד הכותב! האחריות על השימוש בתוכן זה חלה כולה על המשתמש בו! ע"י רונן גיל . הרישיון לשימוש במסמך זה כפוף לתנאי הרישיון המופיעים מטה. חלק מהזכויות שמורות.
מסמך זה מופץ תחת רישיון ייחוס, שימוש לא מסחרי, איסור יצירות נגזרות 2.5 ישראל

פונקציית פייתון להמרת ערכי RGB לערכי HSV*:

```
def RGB2HSV(R=0, G=0, B=0):
   r = R / 255.0 #
   g = G / 255.0 # Reducing rgb values from 8bit to Decimal percentile.
   b = B / 255.0 #

### Evaluating the Minimum and Maximum Values and the difference between them.
   CMin = min(r, g, b)
   CMax = max(r, g, b)
   Delta = CMax - CMin

### Calculating the Value.
   V = round((CMax), 4)

   if Delta == 0: # White or Black or a shade of pure Gray.
      H = 0
   else: # Calculating Hue for all other Colors.
      if CMax == r:
         H = (g-b)/Delta
      elif CMax == g:
         H = (b-r)/Delta + 2
      elif CMax == b:
         H = (r-g)/Delta + 4
      H = int(round(H * 60))
      if H < 0:
         H += 360
   if CMax == 0: # Pure Black.
      S = 0
   else: # Calculating Saturation for all other Colors.
      S = round((Delta / CMax), 4)
      if S < 0:
         S *= (-1)
### Converting Saturation and value to percentages.
   S = str(S * 100) + '%'
   V = str(V * 100) + '%'
   return(H, S, V)
```

* קיימות בפייתון פונקציות דומות, בספרייה colrsys (הסבר מפורט ב https://docs.python.org/2/library/colorsys.html) אך ניתן להשתמש גם בפונקציות אלה כתחלופה. **הערה חשובה:** הקלט והפלט בפונקציות בספריית colorsys הוא תמיד ערכים בין 0 ל 1, בעוד שבפונקציות המופיעות כאן, הקלט הוא תמיד שלושה מספרים בין 0 ל 255 המייצגים, כל-אחד, ערך ב 8ביט של צבע בRGB!

מסמך זה קיים גם בגרסה מקוונת בפורמט HTML בכתובת:
http://ronen.gil.name/Colors/convert.php

מסמך זה נכתב ,ללא שום אחריות מצד הכותב! האחריות על השימוש בתוכן זה חלה כולה על המשתמש בו! ע"י רונן גיל . הרישיון לשימוש במסמך זה כפוף לתנאי הרישיון המופיעים מטה. חלק מהזכויות שמורות.
מסמך זה מופץ תחת רישיון ייחוס, שימוש לא מסחרי, איסור יצירות נגזרות 2.5 ישראל