# מבוא

> הערה: זהו תרגום של ספר המקור בגרסה הזהה לספר [The Rust Programming Language][nsprust] הזמינה בדפוס ובפורמט ebook בהוצאת [No Starch Press][nsp].

ברוכים הבאים לספר \*שפת התכנות ראסט\*, שהינו ספר מבוא לראסט. שפת התכנות ראסט מאפשרת כתיבת תוכנות מהירות ואמינות יותר. כאשר מדובר בעיצוב שפות תכנות, הניסיון לשלב עקרונות תכנות ברמה עילית מחד, ושליטה ברמת סף (low-level control) מאידך, מוביל לרוב להתנגשויות בלתי-פתירות. ראסט קוראת תיגר על קונפליקט זה. ראסט מאזנת יכולת טכנית עצמתית וחווית פיתוח מעולה, ומאפשרת שליטה בפרטי סף (כמו שימוש בזיכרון) ללא צורך בטרחה הרבה ששליטה כזו לרוב גוררת אחריה.

## קהל היעד של ראסט

ראסט היא שפה אידאלית עבור קבוצות משתשמשים שונות, ומסיבות שונות. הבה נתמקד בכמה מהקבוצות היותר חשובות.

### קבוצות מפתחים

ראסט מוכיחה את עצמה ככלי יעיל לשיתוף פעולה בין קבוצות גדולות של מפתחים המביאים עימם רמות שונות של ידע בפיתוח מערכות. קוד ברמת סף הינו רגיש ביותר לבאגים חמקמקים, שברוב שפות התכנות מתגלים רק באמצעות בדיקות מקיפות וסקירת קוד קפדנית ע"י מפתחים מנוסים. בראסט, לעומת זאת, הקומפיילר מגן על הקוד מפני באגים בכך שהוא מסרב מראש לקמפל קוד העשוי להכיל באגים כאלה, כולל באגים הנובעים לרוב מתכנות מקבילי. באמצעות עבודה יחד עם הקומפיילר, יכול הצוות לבלות את זמנו בהתמקדות בלוגיקה התפעולית של התוכנית במקום בלרדוף אחר באגים.

ראסט גם מביא איתו כלי פיתוח עכשוויים לעולם פיתוח המערכות:

\* קארגו (Cargo), מנהל התלותות (dependency manager) וכלי הבניה (build tool) המופץ כחלק מראסט, הופך את פעולות ההוספה, קימפול, וניהול תלותות לתהליך פשוט ועקבי עבור כל משתמשי האקוסיסטם של ראסט.

\* כלי הפירמוט Rustfmt מוודא אחידות בסגנון הכתיבה של כל המפתחים.

\* שרת הראסט (The Rust Server) מפעיל סביבת פיתוח אינטגרטיבית (IDE) ליישום השלמת קוד (code completion) והודעות שגיאה תלויות שורה (inline error messages).

ע"י שימוש בכלים אלו, ובכלים אחרים ממערכת ראסט בכללותה, יכולים מפתחים להיות פרודוקטיביים בעודם כותבים קוד מערכות.

### סטודנטים

ראסט מתאימה לסטודנטים המעוניינים ללמוד אודות רעיונות הקשורים למערכות. משתמשים רבים למדו על נושאים כמו פיתוח מערכות הפעלה בזכות ראסט. הקהילה סביב ראסט מסבירת פנים עד מאוד, ושמחה לענות על שאלות של תלמידים. דרך יוזמות כמו ספר הזה, צוותי ראסט מעוניינים להפוך נושאים הקשורים לתכנות מערכות לנגישים יותר עבור מספר גדול יותר של אנשים, ובמיוחד עבור אלו העושים את דרכם לראשונה בעולמות התכנות.

### חברות

מאות חברות, גדולות כקטנות, משתמשות בראסט בפיתוח במגוון מטלות, כולל כלי פקודת שורה (CLI), שרותי רשת, כלי DevOps, מכשירים משובצים (embedded devices), ניתוח והמרה של קול ו-וידאו, מטבעות מבוזרים, ביו-אינפורמטיקה, מנועי חיפוש, שימושי מרשתת הדברים (Internet of Things), לימוד מכונה, ואפילו חלקים נכבדים של הדפדפן Firefox.

### מפתחי קוד פתוח

ראסט מיועדת למי שמעוניינים לפתח את שפת התכנות ראסט, להכיר את הקהילה סביבה, כלי מפתחים, וספריות. נשמח מאוד אם גם אתם תתרמו לפרוייקט שפת ראסט.

### משתמשים המעריכים מהירות ויציבות

ראסט מיועדת למי שמשתוקקים למהירות ויציבות בשפת התכנות שלהם. כשאנחנו אומרים "מהירות", אנו מתייחסים הן למהירות הריצה של קוד הכתוב בראסט, והן למהירות כתיבת הקוד עצמו. הקומפיילר של ראסט מבטיח יציבות באמצעות הוספת יכולות וארגון מחדש של הקוד (refactoring). זאת בניגוד לקוד קדמון (legacy code) שנוטה להיות שברירי למדי בשפות נטולות-קומפיילר מעין זה, מצב הגורם למפתחים רבים לחשוש מביצוע שינויים בקוד קיים. ראסט שואפת להפוך קוד בטוח לקוד שהוא גם בטוח וגם מהיר. זאת, על-ידי שאיפה להפשטות (קרי, אסטרקציות) ללא-עלות, ויכולות עיליות המתקמפלות לקוד סף במהירות המשתווה לקוד שנכתב ידנית.

השפה ראסט מקווה לתמוך במשתמשים רבים אחרים; אלה המוזכרים כאן מהווים הם אחדים מהקבוצות המושקעות בפרוייקט. במונחים רחבים יותר, השאיפה המרכזית של ראסט היא לקרוא תיגר ואף למגר את יחסי הגומלין השליליים בהם נאלצו מתכנתים לעבוד זה שנים רבות, המתח שבין תכנון עילי ותכנות סף. ראסט עושה זאת על-ידי יצירת סביבת פיתוח שמספקת בטיחות \*בנוסף\* לפרודוקטיביות, וכן מהירות \*בנוסף\* על נוחות ארגונומית. אתם מוזמנים להתנסות בראסט ולראות אם הבחירות של ראסט מתאימים לכם.

## למי מיועד ספר זה

ספר זה מניח שכבר כתבתם קוד בשפת תכנות כזו או אחרת, אבל אינו עושה הנחות לגבי איזו שפת תכנות זו. עשינו ניסיון להציג את החומר בצורה שתהיה נגישה לקהל קוראים בעל רקע מגוון בעולם התכנות. אנחנו לא נשקיע זמן רב בלדון \*מהו\* תכנות או כיצד לחשוב על תכנות. אם אתם חדשים לחלוטין בעולם התכנות, היינו ממליצים על בחירת ספר המכיל הקדמה כללית לעולם התכנות.

## כיצד להשתמש בספר זה

באופן כללי, ספר זה מניח שאתם מתעתדים לקרוא אותו בסדר בו הוא כתוב, מתחילתו ועד סופו. פרקים מאוחרים יותר מסתמכים על מושגים מפרקים מוקדמים יותר, ופרקים מוקדמים עשויים לא לנבור לעומק הפרטים של נושא זה או אחר, כיוון שאלו יידונו מחדש בפרק מאוחר יותר.

בספר זה תמצאו שני סוגי פרקים: פרקי מושגים ופרקי פרוייקטים. בפרקי מושגים תלמדו היבטים שונים של ראסט כשפה. בפרקי פרוייקטים תבנו ותקבצו תכניות קטנות יחדיו, ותיישמו את שלמדתם עד אותו הפרק. פרקים 2, 12, ו-20 הם פרקי פרוייקטים; שאר הפרקים הם פרקי מושגים.

פרק 1 מסביר כיצד להתקין את ראסט, איך לכתוב תוכנית "!Hello, world", ואיך להשתמש בקארגו, מנהל החבילות וכלי הבניה של ראסט. פרק 2 מהווה הקדמה פרקטית לכתיבת תוכנית בראסט, בו תבנו משחק ניחוש מספר. בפרק הזה נסקור מושגים ברמה עילית, ובפרקים מאוחרים יותר נספק פרטים רלוונטיים נוספים. אם אתם מעוניינים להפשיל שרוולים ולקפוץ ישר למים, פרק 2 הוא המקום לפנות אליו.

פרק סוקר מכסה תכונות של ראסט הדומות לאלו של שפות תכנות אחרות, ובפרק 4 תלמדו על מערכת הבעלות של ראסט. אם אתם למדנים קפדניים במיוחד ומעדיפים ללמוד כל פרט לפני שאתם עוברים הלאה, אתם מוזמנים לדלק על פרק 2 ולפנות ישר לפרק 3, ולחזור לפרק 2 כאשר אתם מוכנים לעבוד על פרוייקט המיישם את הפרטים שתלמדו.

פרק 5 נסוב סביב מבנים ומתודות, ופרק 6 מכסה מבחרים (enums), תבניות `התאמה` (match expressions), ותבנית שליטת הזרימה `if let`. אתם תשתמשו במבנים ומבחרים בבניית טיפוסים משלכם בראסט.

בפרק 7 תלמדו על מערכת המודולים של ראסט ואודות חוקי הפרטיות לארגון קוד, וה-API הפומבי שלה. פרק 8 דן בכמה מבני נתונים מסוג אוספים (collection data structures) נפוצים המסופקים ע"י הספריה הסטנדרטית, בכללם ווקטורים, מחרוזות, ומפות גיבוב (hash maps). פרק 9 מציג את הפילוסופיה והטכניקה של ראסט לטיפול בשגיאות.

פרק 10 נובר לעומקם של טיפוסים גנריים (generics), תכונות (traits), ואורכי-חיים (lifetime), אשר נותנים לכם את הכוח להגדיר קוד המותאם לריבוי טיפוסים. פרק 11 נוגע לבדיקות (testing), נושא שאפילו בנוכחות מנגנוני הבטיחות של ראסט הוא הכרחי על מנת להבטיח את נכונות הלוגיקה התפעולית של תוכניותכם. בפרק 12 נבנה יישום משלנו לתת-קבוצה של היכולות של `grep`, כלי שורת הפקודה לחיפוש בטקסט בתוך קבצים. לשם כך, נשתמש במושגים רבים מאלו שנדונו בפרקים הקודמים.

פרק 13 עוסק בסגורים (closures) ובעוקבים (iterators): תכונות של ראסט המגיעות מעולם שפות התכנות הפונקציונאליות. בפרק 14 נתבונן יותר לעומק בקארגו וניגע בעקרונות המומלצים לשיתוף ספריות עם מפתחים אחרים. פרק 15 מדבר על מצביעים חכמים (smart pointers) שהספריה הסטנדרטית מספקת ובתכונות שמאפשרות את תפקודם.

בפרק 16 נעבור על כמה מודלים של תכנות מקבילי ונדבר על כיצד ראסט מאפשרת תכנות מרובה פתילים (multiple threads) חסר-פחד. פרק 17 סוקר את הדומה והשונה שבין אידיאומות ראסט (Rust idioms) לעקרונות תכנות מוכוון אובייקטים (object-oriented programming) שעשויים להיות מוכרים לכם משפות תכנות אחרות.

פרק 18 סוקר דפוסים והתאמת דפוסים (patterns and pattern matching), שהן דרכים עוצמתיות לביטוי רעיונות בתוכנות ראסט. פרק 19 מכיל פסיפס של נושאים מתקדמים, כולל ראסט לא בטיחותי, מאקרואים, ועוד אודות אורכי-חיים, תכונות, טיפוסים, פונקציות, וסגורים.

בפרק 20 נעבוד על פרוייקט בו ניישם ברמת סף שרת רשת מרובה תהליכונים!

לבסוף, בכמה נספחים תמצאו מידע שימושי על השפה בפורמט נוח להפניות. נספח א' מכסה את מילות המפתח של ראסט, נספח ב' מכסה את האופרטורים והסימונים של ראסט, נספח ג' מכסה תכונות נגזרות הנמצאות בספריה הסטנדרטית, נספח ד' מכסה כמה כלי פיתוח שימושיים, ונספח ה' מסביר על מהדורות ראסט השונות. בנספח ו' תוכלו למצוא תרגומים של הספר, ובנספח ז' נגלה איך ראסט נבנית, ומהי ראסט לילית (nightly Rust).

אין דרך שגויה לקרוא את הספר הזה: אם אתם רוצים לקפוץ קדימה, לכו על זה! אתם עלולים להצטרך לקפוץ בחזרה לפרקים מוקדמים יותר במידה ואתם נתקלים בדברים שאינם ברורים. אבל, עשו את מה שעובד עבורכם.

<span id="ferris"></span>

חלק חשוב בתהליך למידת ראסט הוא לימוד כיצד לקרוא את הודעות השגיאה שהקומפיילר מציג: אלה ינחו אתכם בבטחה לעבר קוד שעובד. לכן, נספק דוגמאות רבות שאינן עוברות קומפילציה ביחד עם הודעות השגיאה מהקומפיילר. זכרו שאם אתם מקלידים את הקוד של דוגמא אקראית, יכול להיות שהיא לא תעבור קומפילציה! וודאו שאתם קוראים את הטקסט מסביב לדוגמא על מנת לראות האם הדוגמא בה אתם מתעניינים אמורה להדגים שגיאה. פריס הסרטן (Ferris) יעזור לכם לזהות קוד שלא אמור לעבוד:

| פריס | משמעות |

| ---------------------------------------------------------------------------------------------------------------------------------- | -------------------------------- |

| <img src="img/ferris/does\_not\_compile.svg" class="ferris-explain" alt="Ferris with a question mark" /> | קוד זה לא עובר קומפילציה! |

| <img src="img/ferris/panics.svg" class="ferris-explain" alt="Ferris throwing up their hands" /> | קוד זה נכנס לפאניקה! |

| <img src="img/ferris/not\_desired\_behavior.svg" class="ferris-explain" alt="Ferris with one claw up, shrugging" /> | קוד זה לא מבצע את הפעולה הרצויה. |

ברוב המקרים ננחה אתכם לכיוון הגרסה המתוקנת של כל קוד שאינו עובר קומפילציה.

## קוד מקור

קבצי המקור מהם ספר זה הופק נמצאים ב-[GitHub][book].

[nsprust]: https://nostarch.com/rust

[nsp]: https://nostarch.com/

[book]: https://github.com/IttayWeiss/rustbook-heb