



סקריפט לבדיקת רווחים

נוחות השימוש בסקריפט היא מאוד גבוהה כך שכמעט ואין צורך להסביר את אופן השימוש בו, כיון שטרחנו ליצור ממשק עבודה נוח ופשוט. ורק נגדיר בקצרה כיצד הסקריפט מבצע את הבדיקות.

בחלק העליון של הסקריפט מגדירים מה הסקריפט צריך לבדוק, ישנם 11 סימונים הניתנים לבדיקה. אלא שלגבי כל סימן מבצע הסקריפט בדיקה שונה על פי הצורך הספציפי של אותו סימן. כי לפני סימון של נקודה אסור שיהיה רווח, ולכן הסקריפט יבדוק "האם יש רווח מיותר לפני הנקודה", מאידך לפני סוגר פותח חייב שיהיה רווח ולכן הסקריפט יבדוק "האם חסר רווח לפני הסוגר".

באופן כללי יש 4 **בדיקות** אפשריות, כפי הצורך של התו הנבדק (ראה טבלה בסוף).

תחילה הסקריפט יוצא לחפש את אחד מהסימנים שהוגדרו. בעת שמצא סימן הצריך בדיקה, מבצע הסקריפט את הבדיקות המתאימות כפי המתואר בטבלה. כך שבעת שמצא סוף פסקה מבצע בדיקה האם יש רווח מיותר לפני. ואם מצא סוגר פותח מבצע בדיקה האם חסר רווח לפני, ובאם לא חסר מבצע בדיקה נוספת האם יש רווח מיותר אחרי. הכל כפי מה שמובא בטבלה לעיל.

בסקריפט כבר מוגדר באופן טבעי שלא להתייחס לכל התוים הנסתרים של תוכנת תג, דהיינו לכל אותם תוים הנכנסים תחת הכותרת של אסקי 19 (קודים, סימוני קישורי הערות) וכן לסימנים כמו הוסף גוף, גוף 0 ועוד. דהיינו שהסקריפט מתייחס אליהם בבדיקה כאילו אינם נמצאים, שהרי לפי האמת אינם נראים בעת ההדפסה.

ולכן כאשר הסקריפט בודק "האם יש רווח מיותר לפני סוגר", הוא כלל לא מתייחס לקוד שיש שם לפני או כמה קודים הנמצאים שם, וכן לגבי סימוני קישורי הערות.

התעלם מרצף של 3 נקודות ...

ישנה אופציה של "התעלם מרצף של 3 נקודות". אם מוצא רצף של 3 נקודות אינו בודק כלל אם יש רווח לפני או רווח אחרי, אלא פשוט מדלג הלאה. אם מוצא רצף של יותר מ-3 נקודות, שואל האם למחוק את הנקודות היתרות.

הגדרות של סימוני סוף פסקה

מלבד מה שהסקריפט בודק האם יש רווח מיותר לפני סוף פסקה, ישנה אפשרות לבדיקה האם חסר נקודה או נקודותיים לפני סוף פסקה.

נכנסים להגדרות ומגדירים שם האם לבדוק חסרון נקודה בסוף פסקה. ישנה אפשרות להתעלם מפסקאות שמתחילות בקודים מסויימים (כגון כותרות) שאז לא אמורה להיות נקודה. יש להפריד את הקודים על ידי פסיק. כמו כן ישנה אפשרות להתעלם מפסקאות שמופיע סימן פרק מסויים בשורה הראשונה.

חפש רווחים

בס"ד

בדיקות:

בדוק רווחים כפולים (בלבד)

הגדרות סימוני סוף פסקה

סוגריים

עגול פותח (עגול סוגר)

מרובע פותח (מרובע סוגר)

סימוני פיסוק

נקודה . נקודותיים :

פסיק , נקודה פסיק :

סימן שאלה ? סימן קריאה !

סימנים בבדיקה:

.....

הגדרות

רווח קבוע נחשב כרווח

רווח TS,EN,EM נחשב כרווח

רווח HS נחשב כרווח

רווח m מיקרו נחשב כרווח

ציון דף גמרא נחשב בסדר ()

נקודותיים גם בסוף פסקה:

לא לדווח תקלה אם יש מוביל

לא לדווח תקלה אם יש גרפיקה

לא לדווח תקלה אם יש הרמה

הגדרות רצף הנחשב בסדר

התעלם מרצף של 3 נק'...

בעת מציאת תקלה:

הצג אפשרות תיקון

עצור, ואל תציע תיקונים

הגדרות סוף פסקה

בדוק חסרון נקודה לפני

או חסרון נקודותיים לפני

התעלם מפסקאות:

עם קוד בתחילת פסקה:

עם פרק בשורה הראשונה:



הסבר קצר לגבי הגדרת רצף תוים הנחשב בסדר

ישנה אפשרות של הגדרת רצף תוים הנחשב בסדר (יש תחילה להבין איך זה עובד, אחרת אפשר לטעות בזה):

ההסבר לשימוש בזה הוא כזה:

יש להכניס לרצף מינימום שני תוים, והסקריפט מתייחס לתו הראשון או לתו האחרון ומתעלם משאר התוים בזה האופן:

1. אם התו הראשון שהוכנס הוא מהתוים שנבדקים ע"י הסקריפט, אז הוא נכנס לזכרון של תו רצף, וכל מה שהוכנס אחריו נכנס לזכרון של התעלמות:

- **למשל:** אם כתבתי `@!?` אז התו הראשון שהוא `!` הוא תו הרצף והתוים שאח"כ `@!` הם תוי התעלמות.

דהיינו שכל פעם שהסקריפט ימצא `!`, הוא ילך לחפש רווח אחרי ה-`!` תוך התעלמות מכל התוים האלה `@!`

2. וכן הוא הפוך, אם התו האחרון הוא מהתוים שבבדיקה, אז הוא נכנס לזכרון תו רצף, וכל מה שהוכנס לפניו נכנס לזכרון של התעלמות:

- **למשל:** אם כתבתי `!$@!` אז התו האחרון שהוא `!` הוא תו הרצף והתוים שלפניו `!$@!` הם תוי התעלמות.

דהיינו שכל פעם שהסקריפט ימצא `!`, וילך לחפש רווח לפני ה-`!` הוא יתעלם מכל התוים האלה `!$@!`

3. כשאתה מגדיר רצף מסויים אתה חייב שהתו הראשון או התו האחרון יהיו מהתוים הנמצאים בבדיקה.

- **לדוגמא:** אם תגדיר רצף כזה `@&` אין לא שום משמעות. אבל רצף כזה `@(` או רצף כזה `)@` זה בסדר.

4. כשאתה מגדיר רצף מסויים פירושו שאתה אומר לסקריפט שמצב כזה הוא בסדר ואינו תקלה.

- **לדוגמא:** אם בטקסט הכנסתי אחרי כל סוגר עגול סימון `@` לצורך מעקב וזה נראה כך `(שלום רב)@`.

אז לכאורה יש כאן סוגר שאין אחריו רווח והסקריפט ידווח תקלה.

לכן אני אגדיר בתור רצף הנחשב בסדר את הסימנים `@(`. ואז הסקריפט מסתכל כאילו אין את הסימון `@` אחרי הסוגר והולך לחפש האם יש רווח לאחרי כן.

וממילא אם יהיה כתוב `(שלום רב)@` לאוהבי תורתך. הרי שהסקריפט יתחשב שזה בסדר.

אבל אם יהיה כתוב `(שלום רב)@לאוהבי תורתך`. הרי שחסר רווח אחרי הסוגר – כי הסקריפט התעלם מה-`@` ובדק מה קורה בהמשך.

5. ניתן להגדיר רצף של יותר מ-2 תוים, אבל יש להבין איך זה עובד:

- **לדוגמא:** אם יש לי אחרי סוגר סימון `@13` וזה נראה כך `(שלום רב)@13`

אז יהיה צריך להגדיר רצף כזה `@(13` ואז הסקריפט יחפש לראות האם חסר רווח אחרי ה-`@13`. אבל ממילא גם בכלל זה כעת מוגדר להתעלם מ-`@(1` או `@3`

6. ניתן להגדיר 5 סוגי רצף שונים, כך שאם יש ... וגם `!!!` אז נגדיר רצף אחד .. (מספיק שתי נקודות, כנ"ל) ורצף שני `!?`.

7. אם יש רצף `!!!!` אז נצטרך להגדיר רצף אחד `!!` ורצף אחד `!?`. וכן על זה הדרך.

ב"ה שסייע בעדנו גם בזה, רק אבקש שאם יש איזה שהיא טעות או איזה שהוא שיפור או הסבר שחסר – ליצור עימי קשר ולהודיע. כל שכן אם יש איזה פרט בסקריפט שצריך שיפור או תיקון. תודה.



טבלת האפשרויות חיפוש

עקוב אחר הטבלה המובאת בזה, ותמצא שבעצם יש חלוקה של תוי הבדיקה ל-4 קבוצות של סוגי בדיקה.

קבוצה	תוי בדיקה	1. בדוק אם חסר רווח לפני	2. בדוק אם יש רווח מיותר לפני	3. בדוק אם חסר רווח אחרי	4. בדוק אם יש רווח מיותר אחרי	הערות
1	סוף פסקה	x	*✓	x	x	*אבל אולי זה סוף שורה ששייך לסוף פסקה הקודם כגון שיש פסקה ריקה.
)	✓	x	x	✓	
2]	✓	x	x	✓	
	(x	✓	*✓	x	*אבל אולי סוף פסקה
3	[x	✓	*✓	x	*אבל אולי סוף פסקה
	?	x	✓	*✓	x	*אבל אולי סוף פסקה
	!	x	✓	*✓	x	*אבל אולי סוף פסקה
	.	x	✓	*✓	x	*אבל אולי סוף פסקה
	:	x	✓	*✓	x	*אבל אולי סוף פסקה
	,	x	✓	✓	✓	x
4	,	x	✓	✓	x	
	;	x	✓	✓	x	