- הכל מתבצע סדרתי, אז פעולות שלוקחות זמן נקרא להן blocking כי הן תוקעות לנו את המחסנית
- נגיד שיש לנו פונקציה סנכרונית שאנחנו מחכים שתסתיים ולוקח הרבה מאוד זמן
    - אז לזה אננו קוראים לחסום אתהcallstack כי אנחנו מחכים שזה יסתיים עורק אז נוכל להמשיך
- הדפדפדן או הcpp של הNode הם כן ט'רדים מרובים
    - אני יכול לשלוח אליהם פונקציות שהם ישימו לי בתור לביצוע וכשאוכל, אגיע אליהן.
        - גאוני!
        - ככה אני מסיים את כל הקוד שלי בjs ואז אני מתפנה כדי לדברים שנכנסים לי מהרשת או מהnode
        - ובגלל זה לא באמת משנה באיזה סדר אני מכניס איוונטים כי התשובות נכנסות לי אל התור באופן עצמוני כשהן מסתיימות, אז מי שסיים ראשון יופיע ראשון
- הרינדור של מה שאני רואה יושב במקביל לcallback queue ואנחנו שולפים משם פריימים אם סיימנו להריץ את הjs
    - לכן אם אני תוקע את המחסנית כל האתר שלי יתקע

בaplicitive order קודם מחושבים ערכים מארגומנטים ורק לאחר מכן נעשת אבלוציה של הbody, לכן כדי לעשות את ההחלפה צריך להחזיר את הvalue לCexp.
לעומת זאת בnormal order חישוב הארגומנטים "נדחה" ובעצם לא מתבצע לפני חישוב ה'body לכן אין צורך להשתמש בvalueToLit כי הארגומנטים הנתונים עדיין בצורת Cexp