

نمونه‌سازی اولیه چیست و چرا در دنیای نرم‌افزار اهمیت دارد؟



نمونه‌سازی اولیه به مدلی اشاره دارد که در آن یک نمونه اولیه از یک برنامه کاربردی یا محصول ساخته می‌شود. نمونه‌ای که ویژگی‌های کلیدی محصولی را که در دست طراحی است، در خود جای داده، اما در نظر ندارد منطق اصلی برنامه را نشان دهد.

اگر با دنیای کسب‌وکار استارت‌آپ آشنایی داشته باشید به خوبی می‌دانید در فرهنگ استارت‌آپی مفهومی به نام کمینه محصول پذیرفتنی وجود دارد. کمینه محصول پذیرفتنی یا به اختصار MVP نسخه‌ای خاص از یک محصول است که به تیم طراحی کننده اجازه می‌دهد تا حداکثر یادگیری معتبر (Validated Learning) در ارتباط با مشتریان و نیازهایشان را با کمترین زحمت ممکن به دست آورند. به عبارت دیگر MVP سریع‌ترین راهکار برای پیمودن چرخه بازخوردی بساز/ اندازه بگیر/ یاد بگیر (Build-Measure-Learn) را با کمترین زحمت ممکن پیش روی تیم‌های استارت‌آپی قرار می‌دهد. MVP درست در نقطه مقابل فرآیند سنتی ساخت یک محصول قرار دارد. جایی که فرآیندها زمان‌بر بوده و شامل رشد محصول از آغاز تا رسیدن به یک محصول نهایی می‌شوند. به طور خلاصه MVP سرآغازی بر فرآیند یادگیری و خاتمه دادن به این فرآیند است. در مهندسی نرم‌افزار مفهومی مشابه با MVP به نام **نمونه‌سازی اولیه (Prototype)** وجود دارد. نمونه‌ای که ویژگی‌های کلیدی محصولی را که در دست طراحی است، در خود جای داده، اما در نظر ندارد منطق اصلی برنامه را نشان دهد.

نمونه‌سازی اولیه یک نرم‌افزار، به یکی از محبوب‌ترین پارادایم‌های توسعه نرم‌افزاری تبدیل شده است. به دلیل این که به شرکت‌ها اجازه می‌دهد در همان فازهای اولیه توسعه یک محصول نیازهای مشتری را درک کنند. **نمونه‌سازی اولیه** به طراحان نرم‌افزار و توسعه‌دهندگان اجازه می‌دهد بازخوردهای مشتریان را دریافت کرده و به درستی اطلاع پیدا کنند مشتریان از محصولی که در دست ساخت است چه انتظاراتی دارند. به بیان ساده، نمونه اولیه، یک مدل کاری از یک نرم‌افزار بوده که همراه با یکسری قابلیت‌های محدود منتشر می‌شود. به عبارت دیگر، قرار نیست همراه با ارائه یک نمونه اولیه از یک نرم‌افزار الگو و منطق تجاری یک نرم‌افزار نیز نشان داده شود. با این حال، نمونه اولیه ضمن آن که به کاربران اجازه می‌دهد طرح‌های اولیه طراحان را پیش از آن که محصول به شکل کامل پیاده‌سازی شود ارزیابی کنند، به خود طراحان نیز این قابلیت را می‌دهد تا نیازهای خاص کاربران را به درستی درک کنند. نیازهایی که ممکن است در فاز طراحی محصول مورد بررسی قرار نگرفته باشند. شاید بیلدهای مختلفی که از ویندوز 10 منتشر می‌شوند نمونه قابل تاملی در این زمینه باشند. دانیل رامهال، طراح معروف نرم‌افزار می‌گوید: «به عنوان یک طراح به این نکته دقت کنید که پیش از آغاز به کار و قبل از آن که فاز اجرایی یک پروژه را شروع کنید، بهتر است ایده‌های خود را آزمایش کنید تا مطمئن شوید، موثرترین راهکار برای حل مسئله و انجام کارها را انتخاب کرده‌اید.

نمونه‌سازی اولیه راهکاری برای پاسخ‌گویی به این پرسش است که بهترین روش‌ها و مناسب‌ترین تکنیک‌ها برای به سرانجام رساندن یک پروژه چیست. مهم نیست پروژه‌ای را که در نظر دارید به سرانجام برسانید، یک وب‌سایت یا

یک ماژول نرم‌افزاری باشد، در ابتدا بهتر است ایده‌های اولیه خود را آزمایش کنید تا اطمینان حاصل کنید آیا ابزارها و راهکارهای موجود با هدف شما هماهنگی دارند یا خیر. نمونه اولیه در حقیقت یک نسخه اولیه یا ابتدایی است که سایر ملزومات و مولفه‌ها بر مبنای آن توسعه داده شده و تعریف می‌شوند. این نمونه اولیه دیدگاهی کلی در ارتباط با طراحی و هر نوع تغییر موردنیاز ارائه می‌کند تا به این ترتیب عملکردتان در طراحی بهبود پیدا کرده و پروژه نهایی به مطلوب‌ترین شکل به سرانجام برساند. برای این منظور به نرم‌افزارها و سرویس‌هایی نیاز دارید تا در ساخت یک نمونه اولیه به شما کمک کنند.»

کاربران نهایی و افرادی که به آن‌ها آزمایش‌کنندگان (tester) گفته می‌شود، قادر هستند نمونه اولیه را آزمایش کرده و در ادامه بازخوردهای خود را در ارتباط با نیازمندی‌هایی که در نرم‌افزار قرار نگرفته، مشخص کرده و به تیم سازنده ارائه کنند. رویکردی که اینسایدرهای ویندوز 10 این کار را انجام می‌دهند. نمونه‌سازی اولیه درست نقطه مقابل چرخه توسعه یکپارچه است که در دهه 60 و 70 میلادی برای ساخت یک موجودیت کامل از یک برنامه در ابتدا و کشف تناقضات مابین طراحی و پیاده‌سازی در مرحله بعد که در نهایت به افزایش هزینه‌های نرم‌افزار و برآورد ضعیفی از زمان و هزینه‌ها منجر می‌شد، ارائه شد. در روش یکپارچه، فرض بر این است که طراح نرم‌افزار و توسعه‌دهنده فهرمانان اصلی هستند که به تنهایی قادر هستند مشکلات را برطرف کنند. نمونه‌سازی اولیه به شرطی که درست به کار گرفته شود دو مزیت مهم به همراه می‌آورد. اول آن‌که مانع شکل‌گیری هزینه‌های سنگین شده و دوم آن‌که اجازه نمی‌دهد در فازهای پایانی تغییرات ناخواسته یا غیرقابل‌پیش‌بینی در نرم‌افزار به وجود آید. تغییراتی که اعمال آن‌ها در محصول نهایی به سختی امکان‌پذیر خواهد بود. توجه داشته باشید که یک نمونه اولیه ساده ممکن است فقط برخی از جنبه‌ها را نشان دهد و همواره این احتمال وجود دارد که نسخه نهایی محصول کاملاً متفاوت از نسخه پیش‌تولید باشد. از شناخته‌شده‌ترین نمونه‌های به‌کارگیری این مدل می‌توان به نمونه‌سازی اولیه مفسر NYU's Ada/ED که برای زبان برنامه‌نویسی Ada پیاده‌سازی شده، اشاره کرد. (Ada یک زبان برنامه‌نویسی است که بر مبنای زبان برنامه‌نویس پاسکال توسعه پیدا کرده است.)

مطلب پیشنهادی



چرا مهندسی نیازمندی‌های نرم‌افزار مهم است؟

نمای کلی فرآیند نمونه‌سازی اولیه

برای آن‌که بتوانید یک نمونه اولیه از یک محصول نرم‌افزاری را تولید کنید، لازم است چهار مرحله‌ای را که در ادامه با آن‌ها آشنا خواهید شد، تکمیل کنید.

1. شناسایی نیازمندی‌های اصلی (Basic Requirement Identification)

در این مرحله باید نیازهای بنیادین نرم‌افزار شناسایی شوند. به‌ویژه ملزوماتی که در تعامل با رابط کاربری هستند. در این مرحله می‌توان از جزئیات پیچیده‌تر مرتبط با طراحی داخلی و همچنین جزئیاتی که در ارتباط با عملکرد و امنیت هستند، صرف‌نظر کرد.

2. طراحی نمونه اولیه (Developing the initial Prototype)

نمونه اولیه‌ای که در این مرحله توسعه داده می‌شود در ارتباط با نیازهای اساسی مرتبط با رابط کاربری و البته مولفه‌های پایه‌ای محصول است. توجه داشته باشید ویژگی‌هایی که در نمونه اولیه قرار می‌گیرند ممکن است در نرم‌افزار واقعی به همین شکل کار نکنند و تغییراتی در آن‌ها اعمال شود. نمونه اولیه ساخته‌شده سعی می‌کند به مشتری دورنمایی کلی از محصولی که طراحی خواهد شد، ارائه کند.

3. بررسی نمونه اولیه (Review of the Prototype)

در این مرحله نمونه اولیه طراحی‌شده به مشتری یا سایر ذی‌نفعان نشان داده‌شده و در ادامه بازخوردها به شکل سازمان‌دهی شده برای بهبود محصول در حال توسعه استفاده می‌شود.

4. بازبینی و بهبود نمونه اولیه (Revise and Enhance the Prototype)

از طریق بازخوردهای به دست آمده از سایر ذینفعان، تغییراتی در نمونه اولیه به وجود خواهد آمد. در این مرحله با مشتریان در ارتباط با عواملی همچون محدودیت‌های زمانی، بودجه و قابلیت‌های فنی که باید به شکل واقعی در محصول پیاده‌سازی شود مذاکراتی انجام می‌شود. تغییرات مورد تایید در نمونه اولیه جدید اعمال شده و چرخه تکرار دومرتبه از سر گرفته می‌شود تا در نهایت انتظارات مشتریان برآورده شود. اگر تغییراتی در نمونه اولیه اعمال شد، مراحل سه و چهار دوباره تکرار می‌شود.

ابعاد نمونه‌سازی اولیه

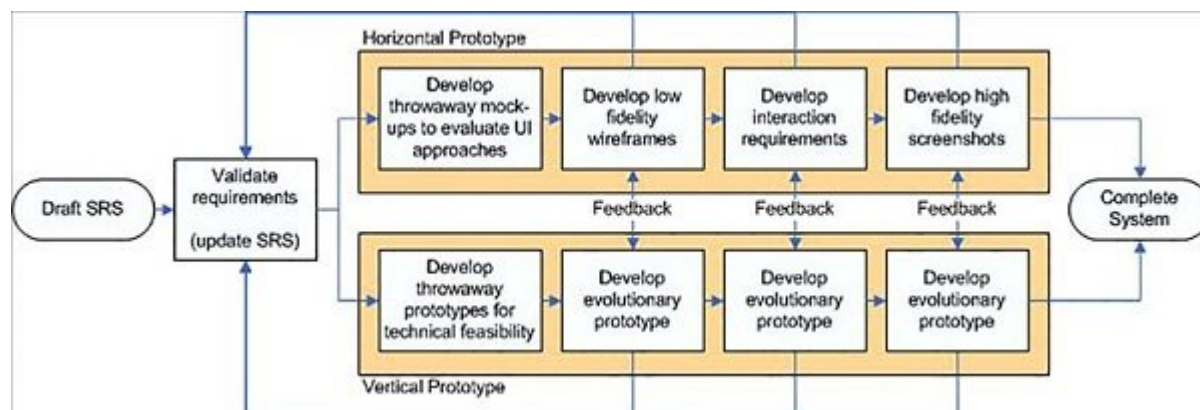
نمونه‌های اولیه را می‌توان به دو شکل افقی و عمودی آماده کرد. نیلسن، در کتاب «مهندسی نرم‌افزار» از دو اصطلاح نمونه‌سازی افقی و عمودی یاد کرده که هر یک از این نمونه‌ها برای اهداف خاصی ساخته می‌شوند.

نمونه‌سازی افقی

نمونه اولیه افقی ضمن آن‌که رابط کاربری محصول را نشان می‌دهد، دورنمایی کلی از ماهیت سیستم را نشان داده، بدون آن‌که روی عملکرد داخلی متمرکز شود. این نمونه اولیه می‌تواند کل پیکره یک سیستم را همراه با زیرسیستم‌های موجود در آن نشان دهد. در نتیجه به‌جای آن‌که به شکل دقیق روی عملکرد کاربردی یک سیستم همچون نحوه دسترسی به بانک‌های اطلاعاتی متمرکز باشد بر تعامل با کاربر تاکید دارد. ما از نمونه‌سازی افقی زمانی استفاده می‌کنیم که در نظر داشته باشیم تاییدیه مربوط به رابط کاربری را به دست آورده و تخمینی اولیه در ارتباط با زمان موردنیاز برای توسعه و هزینه به دست آوریم. نمونه‌سازی افقی و عمودی دو هدف متفاوت از یکدیگر را دنبال می‌کنند. نمونه‌های افقی، با هدف دریافت بیشترین اطلاعات در ارتباط با رابط کاربری و نیازمندی‌های تجاری به کار گرفته می‌شوند. برخی از شرکت‌ها از نمونه‌سازی افقی حتی در ارتباط با دمو فروشی که قرار است در بازار انتشار پیدا کند، استفاده می‌کنند.

نمونه‌سازی عمودی

نمونه اولیه عمودی در نقطه مقابل نمونه افقی قرار داشته و به‌تفصیل عملکرد یک مولفه خاص یا زیرسیستمی را که در یک محصول به کار گرفته شده است، تشریح می‌کند. در نمونه‌سازی عمودی سعی بر این است که جزئیات مربوط به عملکرد یک محصول به‌خوبی شرح داده شود. از جمله مزایای مهم نمونه‌سازی عمودی می‌توان به طراحی درست و دقیق بانک‌های اطلاعاتی، به دست آوردن اطلاعاتی در ارتباط با گسترده بودن داده‌ها، شناسایی نیازمندی‌های پیچیده و... اشاره کرد. شکل (1) به‌خوبی نشان می‌دهد که ترکیب این دو مدل با یکدیگر چطور فرآیند تکمیل شدن یک محصول را به سرانجام می‌رسانند. نمونه‌های اولیه عمودی روی مباحث فنی و تکنیکی متمرکز هستند. این نمونه‌ها به‌منظور ارائه اطلاعات دقیقی در ارتباط با عملکرد زیرسیستم‌هایی که در یک محصول به کار گرفته می‌شوند، ساخته می‌شوند. بررسی نیازمندی‌های بانک اطلاعاتی، نحوه تعامل و بارگذاری داده‌ها و پردازش آن‌ها در زیرسیستم‌های فرعی از جمله این موارد است.





دو مفهوم در ظاهر یکسان، اما در باطن متفاوت
مهندسی نرم‌افزار و برنامه‌نویسی چه تفاوت‌هایی با یکدیگر دارند؟

انواع نمونه سازی اولیه

نمونه‌های اولیه افقی یا عمودی برای یک محصول به روش‌های مختلفی ایجاد می‌شوند. درحالی‌که این نمونه‌ها را به روش‌های مختلفی می‌توان ایجاد کرد، با این حال تمامی مدل‌ها در بطن خود به دو نوع نمونه‌های دوراندختنی اولیه (Throwaway Prototyping) و نمونه‌های تکاملی (Evolutionary Prototyping) اشاره دارند.

نمونه‌های دوراندختنی اولیه

نمونه‌های دوراندختنی اولیه که در برخی منابع به نام «نمونه اولیه سریع» یا «نمونه روبه پایان» از آن‌ها نام برده می‌شود، مدل‌هایی هستند که با صرف زمان کم و حداقل تحلیل نیازمندی‌ها ساخته می‌شوند. زمانی که تیم توسعه نیازهای واقعی را درک کرد، این نمونه اولیه را کنار گذاشته و بر مبنای نیازهای واقعی مشتری فرآیند ساخت محصول را آغاز خواهد کرد. توجه داشته باشید نمونه اولیه دوراندختنی تنها به منظور شناسایی ملزومات اولیه به کار گرفته شده و به بخشی از نرم‌افزار نهایی تبدیل نخواهد شد. این مدل به تیم توسعه‌دهنده کمک می‌کند نیازمندی‌های اولیه را جمع‌آوری کرده و در ادامه یک مدل کاری ساده از سیستم ایجاد کرده تا به شکل بصری به مشتری نشان دهد محصولی که قرار است ساخته شود منطبق با نیازهای او خواهد بود. نکته قابل توجه این‌که مدل کاملاً غیررسمی بوده و سرعت مهم‌ترین فاکتوری است که در زمان ساخت این مدل باید به آن دقت کنید. یک مدل دوراندختنی در حقیقت نقطه شروعی است که به مشتری اجازه می‌دهد یکبار دیگر نیازمندی‌های خود را بازبینی کند. زمانی که نیازمندی‌ها به شکل دقیقی شناسایی شدند این نمونه اولیه کنار گذاشته شده و سیستم به‌طور رسمی بر مبنای ملزومات شناسایی شده ساخته خواهد شد.

نمونه‌سازی تکاملی

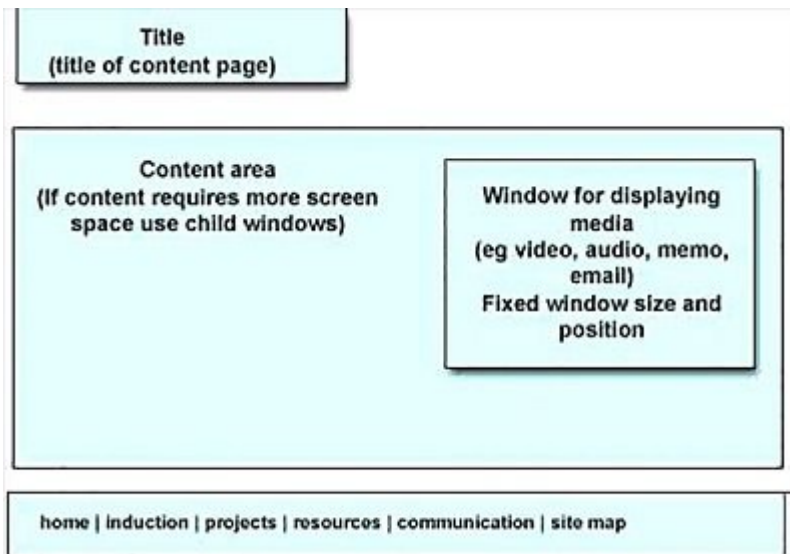
نمونه‌سازی تکاملی که در برخی منابع به نمونه اولیه نگاشت شده نیز معروف است رویکردی متفاوت از نمونه‌سازی دوراندختنی اولیه دارد. نمونه‌سازی تکاملی سعی می‌کند یک نمونه کاربردی واقعی را بر مبنای حداقل عملکرد یک محصول ارائه کند. محصولی که فرآیند ساخت آن تازه آغاز شده است. زمانی که فرآیند توسعه محصول بر مبنای نمونه‌سازی تکاملی انجام می‌شود، سیستم به‌طور مداوم پالایش و بازسازی می‌شود. نمونه‌سازی تکاملی به این مسئله اذعان دارد که ما همه نیازمندی‌ها را نمی‌توانیم درک کنیم و فقط می‌توانیم مولفه‌هایی را ایجاد کنیم که درک درستی از آن‌ها به دست آورده‌ایم.

نمونه‌سازی افزایشی

نمونه اولیه افزایشی به ساخت چند نمونه اولیه کاربردی از زیرسیستم‌ها و سپس ادغام کردن همه نمونه‌های اولیه در دسترس به‌منظور ساخت یک سیستم کامل اشاره دارد.

نمونه‌سازی مفرط

نمونه‌سازی مفرط در ارتباط با توسعه نرم‌افزارهای تحت وب استفاده می‌شود. نمونه‌سازی مفرط شامل سه مرحله متوالی است: در مرحله اول، یک نمونه اولیه و ایستا که شامل صفحات موجود است در فرمت HTML است؛ در مرحله دوم، برنامه‌نویسی صفحات انجام شده و داده‌ها از طریق یک لایه شبیه‌سازی خدمات پردازش انجام می‌شوند، در مرحله سوم، سرویس‌ها پیاده‌سازی شده و در قالب نمونه اولیه نهایی با یکدیگر ادغام می‌شوند. (شکل 2)



نمونه‌سازی نرم‌افزاری به چه منظور استفاده می‌شود؟

نمونه‌سازی اولیه نرم‌افزاری بیشتر در ارتباط با توسعه سامانه‌هایی مفید است که سطح بالایی از تعامل با کاربران را دارند. سیستم‌های آنلاین از جمله این موارد هستند. اما در مقابل در ارتباط با نرم‌افزارهایی که حجم پردازش داده‌ها در آن‌ها بالا است و بخش بزرگی از عملیات به شکل داخلی انجام شده و رابط کاربری نقش چندانی در آن‌ها ندارد، بهتر است از **نمونه‌سازی اولیه** استفاده نشود، به واسطه آن‌که سربار اضافی را به چنین پروژه‌هایی تحمیل می‌کند. نکته‌ای که در انتهای این مقاله باید به آن اشاره داشته باشیم این است که **نمونه‌سازی اولیه** از جمله مدل‌هایی است که باید در مورد به‌کارگیری آن جانب احتیاط رعایت شود.

به عبارت دقیق‌تر، اگر به‌کارگیری نمونه اولیه به ایجاد ارزش‌افزوده قابل‌توجهی برای نرم‌افزار نهایی منجر شود باید از این مدل استفاده کرد.

مزایای نمونه‌سازی اولیه

- از مهم‌ترین مزایای **نمونه‌سازی اولیه** می‌توان به موارد زیر اشاره کرد:
- افزایش مشارکت مشتریان حتی قبل از آن‌که فرآیند پیاده‌سازی محصول انجام شود.
 - با توجه به این‌که یک مدل کاری از محصول به نمایش گذاشته می‌شود، مشتریان درک بهتری از محصول در دست توسعه به دست می‌آورند.
 - این مدل زمان طراحی و هزینه‌ها را کاهش می‌دهد. به دلیل این‌که معایب خیلی زود شناسایی می‌شوند.
 - دریافت سریع بازخوردها از کاربران باعث می‌شود تا راه‌حل‌های بهتری ارائه شود.
 - نبود قابلیت‌ها به راحتی قابل‌شناسایی است.
 - **نمونه‌سازی اولیه** برداشت‌های متفاوت مشتریان و طراحان را از میان برده و به هر دو گروه اجازه می‌دهد دید یکسان و مشترکی نسبت به محصول پیدا کنند.
 - این مدل زمانی که امکان تعریف نیازمندا وجود نداشته یا تعریف نشده‌اند، مناسب است.
 - مشتریان برای بهبود نیازها می‌توانند سیستم را آزمایش کنند.

معایب نمونه‌سازی اولیه

- از مهم‌ترین معایب مدل **نمونه‌سازی اولیه** می‌توان به موارد زیر اشاره کرد:
- همواره این احتمال وجود دارد که به دلیل وابستگی بیش‌ازحد به نمونه اولیه فرآیند تحلیل به شکل کافی و دقیقی انجام نشود.
 - مشتریان ممکن است در تشخیص نمونه‌های اولیه و سیستم واقعی دچار مشکل شوند.
 - این روش همان‌گونه که اشاره شد ممکن است پیچیدگی سیستم را افزایش دهد.

توسعه‌دهندگان ممکن است حتی زمانی که به لحاظ فنی امکان‌پذیر نیست، سعی کنند دوباره از نمونه‌های اولیه

موجود برای ساخت سیستم واقعی استفاده کنند. این مشکل از آن جهت رخ می‌دهد که توسعه‌دهندگان زمان زیادی را صرف آماده‌سازی نمونه اولیه می‌کنند و به نمونه اولیه وابسته می‌شوند. در نتیجه این احتمال وجود دارد تا شاهد ساخت سیستمی فاقد معماری باشیم. فراموش نکنید، با توجه به این که نمونه‌های اولیه دوراندختنی به سرعت آماده می‌شوند، ممکن است این ذهنیت نزد کاربران یا مشتریان به وجود آید که محصول نهایی نیز به همان سرعت ساخته خواهد شد.

منبع:

[csebrules](#)
[lauyin](#)
[study](#)
[istqbexamcertification](#)

تاریخ انتشار:
03 دی 1397

نشانی منبع:

<https://www.shabakeh-mag.com/computer-science/14099/%D9%86%D9%85%D9%88%D9%86%D9%87%E2%80%8C%D8%B3%D8%A7%D8%B2%DB%8C-%D8%A7%D9%88%D9%84%DB%8C%D9%87-%DA%86%DB%8C%D8%B3%D8%AA-%D9%88-%DA%86%D8%B1%D8%A7-%D8%AF%D8%B1-%D8%AF%D9%86%DB%8C%D8%A7%DB%8C-%D9%86%D8%B1%D9%85%E2%80%8C%D8%A7%D9%81%D8%B2%D8%A7%D8%B1-%D8%A7%D9%87%D9%85%DB%8C%D8%AA-%D8%AF%D8%A7%D8%B1%D8%AF%D8%9F>