

## مقدمه ای بر PostScript

نویسنده: سید محمد مهدی رشتی

### پکیده:

عصر حاضر، عصر اطلاعات است و انتقال اطلاعات یکی از مسائل لاینفک زندگی بشر امروز است. در ارسال اسناد، یکی از معزلات ارسال اطلاعات مسئله حفظ فرمت ظاهری سند است. یعنی سند در چه فرمتی باشد، تا دریافت کننده این سند را در ظاهر اولیه ببیند. پست اسکریپت زبانی است که در آن برای حل این مسئله راه‌حلهایی ارائه شده است. در این مقاله به معرفی اجمالی این زبان و آشنایی با برخی توابع پر کاربرد آن می‌پردازیم.

### مقدمه:

پست اسکریپت یک زبان برنامه نویسی است که برای رساندن توصیف یک صفحه دلخواه مجازی به پرینتر از آن استفاده می‌شود. این زبان دارای تعداد زیادی عملگر گرافیکی است که می‌توانند در هر ترکیبی قرار بگیرند. همچنین دارای متغیر است و اجازه استفاده ترکیبی از عملگرها را برای ایجاد رویه‌های پیچیده تر به کاربر می‌دهد.

صفحات پست اسکریپت برنامه‌هایی هستند که توسط یک مفسر اجرا می‌شوند. برنامه‌های پست اسکریپت معمولاً توسط برنامه‌های application که روی کامپیوتر دیگری در حال اجرا هستند تولید می‌شوند. به هر حال تعداد بسیاری پرینتر پست اسکریپت، از جمله Apple LaserWriter، یک محیط محاوره‌ای دارند که ممکن است کاربر به طور مستقیم با زبان پست اسکریپت برنامه نویسی کند.

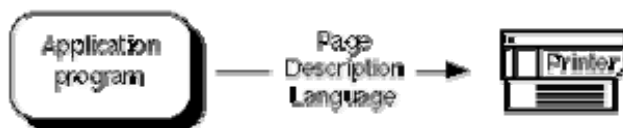
### پست اسکریپت، یک زبان توصیف صفحه:

پست اسکریپت تعداد زیادی توابع گرافیکی دارد که به این زبان اجازه می‌دهند که یک صفحه دلخواه را به صورت مختصر و مفید توصیف کند. این توابع قرار گرفتن ۳ نوع شیء گرافیکی را در صفحه کنترل می‌کنند:

۱- **متن:** متن در فونتهای مختلف، در هر جایی از صفحه، در هر اندازه، در هر جهت و زاویه نسبت به افق می‌تواند قرار بگیرد.

۲- **اشکال هندسی:** توابع گرافیکی پست اسکرپت توانایی ایجاد اشکال هندسی را به کاربر می دهند. این توابع محل قرار گرفتن خطوط مستقیم و منحنی ها را در هر اندازه، جهت، و پهنا، بعلاوه شکل‌های بسته را در هر شکل، اندازه و رنگ توصیف می کنند.

۳- **تصاویر:** هر تصویری در هر جایی از صفحه، با هر اندازه و در هر جهت می تواند قرار بگیرد. تمام اشیاء گرافیکی به راحتی می توانند چرخانده شوند، بزرگ یا کوچک شوند و به بخش‌های مشخص تقسیم شوند.



### مدل تصویری پست اسکرپت:

مدل گرافیکی یک مجموعه از قوانین هستند که برای طراحی یک سیستم گرافیکی جمع شده اند. مدل گرافیکی پست اسکرپت بسیار شبیه مدلی است که ما به طور غریزی برای طراحی با دست استفاده می کنیم. مدل پست اسکرپت یک تصویر را به عنوان فضایی که باید در صفحه با جوهر پرنتر پر شود در نظر می گیرد. جوهر می تواند حروف، خطوط، شکل‌های تو پر، یا قسمتهایی از یک عکس را رسم کند. خود جوهر می تواند سیاه، سفید، رنگی، یا به صورت سایه خاکستری باشد. این عناصر می توانند مرزهای یک تصویر را آن طور که در صفحه قرار گرفته اند، تشکیل دهند. بعد از اینکه صفحه به شکل دلخواه ساخته شد، می تواند توسط پرنتر به چاپ شود.

سه مفهوم اساسی برای پیاده سازی مدل تصویری پست اسکرپت وجود دارند:

۱- **صفحه جاری:** صفحه جاری، صفحه ایده آلی است که پست اسکرپت روی آن رسم می کند. صفحه

جاری مستقل از توانایی های پرنتری است که استفاده می شود.

وقتی برنامه شروع می شود صفحه جاری کاملاً خالی است. توابع ترسیم پست اسکرپت نقطه ها را روی صفحه جاری قرار می دهد، آنهایی که کاملاً تیره هستند در حقیقت نقاطی هستند که روی هم قرار گرفته اند. وقتی که صفحه جاری کاملاً توصیف شد، به پرنتر فرستاده می شود و پرنتر صفحه را با توجه به توانایی خود چاپ می کند.

مهم است به خاطر داشته باشیم که فرقی ندارد که نقطه چه رنگی باشد - سیاه، سفید، خاکستری یا رنگی - رنگ به صفحه جاری واگذارده می شود تا نقطه را چه رنگی کند.

۲- **مسیر جاری:** مجموعه ای از نقاط، خطوط و منحنی های متصل و جدا از هم هستند که در کنار هم شکلها و موقعیت آنها در صفحه را توصیف می کنند. هیچ محدودیتی در مورد شکلهایی که توسط مسیر جاری تعریف می شوند وجود ندارد. اشکال می توانند محدب، مقعر و یا حتی مرکب -چند ضلعی با اضلاع متقاطع- باشند. عناصر مسیر جاری بر حسب موقعیت در صفحه جاری مشخص می گردند. دقت تصویر - resolution - پریتر مورد استفاده به هیچ وجه تعریف مسیر را محدود نمی کند.

خود مسیر جاری یک علامت روی صفحه جاری نیست. عملگرهای مسیر پست اسکرپت مسیر جاری را تعریف می کنند، اما صفحه را علامت گذاری نمی کند. وقتی یک مسیر تعریف شد، این مسیر می تواند روی صفحه جاری نقش شود، - کشیدن یک خط در امتداد مسیر - پر شده و یا اینکه به عنوان یک مرز جدا کننده تصویر استفاده شود.

۳- **مسیر چیده شده:** مسیر جاری چیده شده مرز مساحتی است که باید در آن عمل رسم صورت گیرد. به طور پیش فرض مسیر قسمت کننده مطابق اندازه استاندارد صفحه است. مسیر چیده شده می تواند به هر شکل و اندازه دلخواه تغییر کند. اگر یک عملگر تصویر پست اسکرپت بخواهد در صفحه جاری و خارج از مسیر چیده شده جاری تغییر ایجاد کند، فقط آن قسمت از تغییر که درون مسیر چیده شده جاری قرار می گیرد روی صفحه جاری رسم می گردد.

### سیستم مختصات:

موقعیتها در صفحه به وسیله زوج  $(x,y)$  سیستم مختصات صفحه توصیف می شوند. هر وسیله خروجی یک سیستم مختصات درونی دارد که به وسیله آن نقاط روی یک صفحه را آدرس دهی می کند. ما این سیستم مختصات درونی که برای وسایل مختلف متفاوت است را فضای وسیله می نامیم. فضای وسیله بین پریترها متفاوت است و هیچ استاندارد یکسانی در مورد محل قرار گرفتن مبدا مختصات و مقیاس محورها وجود ندارد.

موقعیتها روی صفحه جاری پست اسکرپت بر حسب سیستم مختصات کاربر یا فضای کاربر توصیف می شوند. این سیستم مختصات مستقل از فضای وسیله پریتر است. مختصات در یک برنامه پست اسکرپت، قبل از چاپ روی صفحه جاری به صورت خودکار از فضای کاربر به فضای پریتر تبدیل می شود. فضای کاربر پست اسکرپت از سه طریق می تواند دچار تغییر شود:

۱- مبدا مختصات سیستم می تواند به هر نقطه در فضای کاربر منتقل شود.

۲- محورهای مختصات می توانند در جهت دلخواه بچرخند

۳- محورها می توانند با درجه دلخواه تغییر مقیاس داشته باشند؛ مقیاس می تواند در مورد محورهای  $x$  و  $y$  متفاوت نیز باشد. یک کاربر می تواند هر تبدیل خطی دلخواه را بین فضای کاربر و فضای وسیله تعیین کند. بنابراین مختصات در یک برنامه پست اسکریپت با توجه به صفحه جاری قابل تغییر است.

### **پست اسکریپت، یک زبان برنامه نویسی:**

حدود یک سوم از مبحث زبان پست اسکریپت مربوط به گرافیک است. مابقی، این زبان را به یک زبان برنامه نویسی عمومی کامپیوتر تبدیل می کند. زبان پست اسکریپت عناصری از زبانهای برنامه نویسی دیگر را در خود جا داده است، اما بیشتر از همه به زبان برنامه نویسی فورث - Forth - مشابهت نزدیکی دارد.

### **پشته پست اسکریپت:**

پست اسکریپت قسمتی از حافظه را برای کار کردن با داده ها به خود اختصاص می دهد که پشته نامیده می شود. پشته مانند یک دسته کتاب روی هم می باشد. آخرین کتابی که روی کتابها قرار می گیرد، اولین کتابی است که باید برداشته شود. به طور مشابه، اعداد، رشته ها، و انواع دیگر داده که روی پشته قرار می گیرند، به ترتیب معکوس از پشته خارج می شوند. آخرین عنصری که به پشته اضافه می شود، اولین عنصری است که از پشته خارج می شود.

### **نمونه پست اسکریپت:**

عملگرهای پست اسکریپت نظیر `add` و `sub` که به اعداد یا انواع دیگری از داده نیاز دارند این داده ها را از پشته می گیرند. برای استفاده از یک عملگر، باید ابتدا داده های مورد نیاز آن عملگر که همان عملوندهای آن هستند را روی پشته بگذاریم و سپس عملگر را فراخوانی کنیم. عملگر نیز نتیجه خاص خود را روی پشته می گذارد. این شیوه برنامه نویسی که عملوندها قبل از عملگر مشخص می شوند، نحو `postfix` نامیده می شود.

### **انواع داده پست اسکریپت:**

پست اسکریپت از بسیاری از انواع داده معمول در سایر زبانها پشتیبانی می کند، نظیر داده `Boolean` ، `Real` ، آرایه ها و رشته ها. زبان پست اسکریپت همچنین انواع شیء نظیر `dictionary` و `mark` را نیز تعریف می کند.

### **انعطاف پذیری پست اسکریپت:**

پست اسکریپت یک زبان بسیار انعطاف پذیر است. توابعی که وجود ندارند، اما مورد نیاز هستند را می شود تعریف کرد و سپس آن را همانند عملگرهای پست اسکریپت استفاده کرد. بنابراین پست اسکریپت زبان ثابتی

نیست که برنامه مورد نیاز را محدود کند، بلکه محیطی است که قابل تغییر برای کار مورد نظر است. قسمتهایی از توصیف یک صفحه می توانند با یکدیگر ترکیب شوند و صفحات پیچیده دیگری ایجاد کنند. این قسمتها می توانند در شکل اصلی خود یا به صورت ترجمه شده، چرخانده شده، و بزرگ یا کوچک شده برای ایجاد هزاران صفحه جدید ترکیبی استفاده شوند.

### **برنامه های قابل چاپ:**

برنامه های پست اسکریپت کاملاً به صورت کارکترهای اسکی قابل چاپ هستند. این ویژگی به برنامه این اجازه را می دهد که مانند فایل های متنی معمولی مدیریت شوند. به علاوه این ویژگی تضمین می کند که یک برنامه پست اسکریپت به سادگی که ساختار برنامه است، برای هر شخص قابل خواندن است.

### **Unicode در پست اسکریپت:**

به خاطر اینکه فایل های پست اسکریپت با کدهای اسکی نوشته می شوند، برای نوشتن با حروفی که در جدول اسکی قرار ندارند با مشکل مواجه هستیم. البته در پست اسکریپت توانایی ایجاد کارکترهای جدید به وسیله CMap وجود دارد، اما کار کردن با آن به آسانی نوشتن کارکترهای اسکی نیست.

در حقیقت به وسیله CMap ما می توانیم تمامی کارکترهای Unicode را به متن پست اسکریپت اضافه کنیم، اما این کار موجب بالا رفتن حجم فایل پست اسکریپت می شود، زیرا باید جدول CMap در این فایل یا فایل های کمکی دیگر نگهداری شود. متأسفانه تا کنون در زمینه فارسی نویسی در پست اسکریپت تلاش چندانی صورت نگرفته است. البته به وسیله Adobe Distiller می توان هر متن را به PDF تبدیل کرد و تبدیل کردن PDF به پست اسکریپت نیز توسط بسیاری از برنامه ها صورت می گیرد. اما در زمینه پست اسکریپت فارسی هنوز فقدان فونتهای فارسی مناسب برای پست اسکریپت وجود دارد.

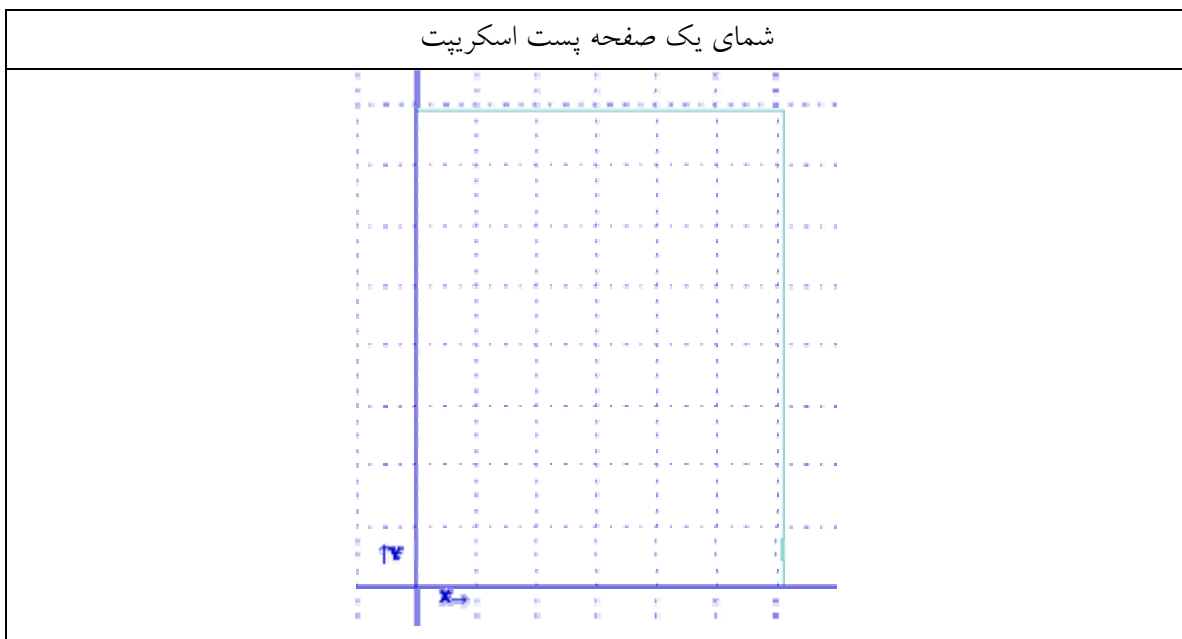
### **آشنایی با تعدادی از عملگرهای پست اسکریپت:**

#### **:Show**

شاید مهمترین عملگر گرافیکی پست اسکریپت، عملگر نمایش متن باشد. در زیر نمونه از طرز استفاده این عملگر را می بینیم:

کد برنامه	اجرای برنامه
/Times-Roman findfont 36 scalefont setfont 144 288 moveto (Hello World.) show showpage	Helo World.

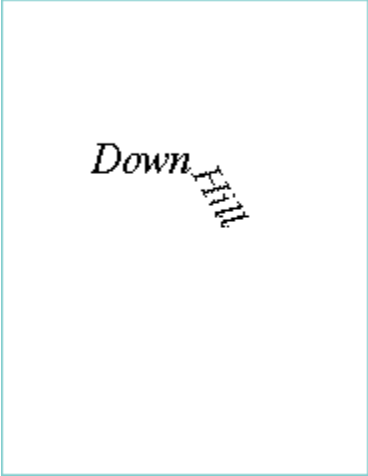
در اولین خط فونت مورد نظر برای نوشتن انتخاب می شود و در خط دوم اندازه قلم تعیین می گردد. در خط سوم محل نوشتن قلم تعیین می گردد. لازم به ذکر است که سیستم مختصات پست اسکریپت به گونه ای است که مبدا مختصات در گوشه پایین و سمت چپ صفحه قرار دارد.



در خط چهارم دستور چاپ متن قرار دارد و دستور خط پنجم باعث می شود که بافر روی صفحه اصلی نقش شود. در حقیقت این دستور برای این است که صفحه درست نمایش داده شود.

## Rotate

این دستور محورهای مختصات را به اندازه زاویه مشخص شده در خلاف جهت عقربه های ساعت می چرخاند. البته اگر زاویه ورودی منفی باشد، جهت چرخش مطابق چرخش عقربه های ساعت است.

کد برنامه	اجرای برنامه
<pre>/Times-Italic findfont 70 scalefont setfont 150 500 moveto (Down) show -60 rotate (Hill) show showpage</pre>	


در این برنامه می بینیم که در خط ششم دستور Rotate قرار دارد و پس از این دستور چاپ کلمه Hill با زاویه ۶۰- درجه نسبت به افق صورت می گیرد.

## Translate

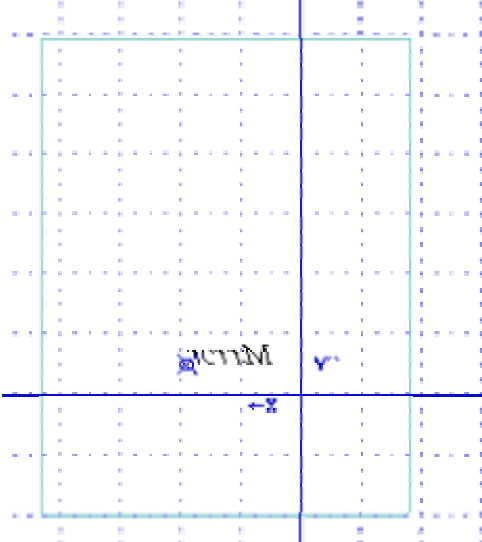
این دستور مبدا مختصات را به نقطه جدید منتقل می کند.

## Scale

این دستور مقیاس محورهای X و Y را به مقیاس جدید تغییر می دهد. به وسیله این دستور می توان به طور مثال در بعد X یا Y یا هر دو تغییر در اندازه متن ایجاد کرد. نمونه ای از کاربرد این دستور را در مثال زیر می بینیم:

کد برنامه	اجرای برنامه
<pre> /Times-Bold findfont 60 scalefont setfont 150 250 moveto 1 5 scale (POSTSCRIPT) show showpage </pre>	

در خط پنجم این برنامه می بینیم که مقیاس محور x ثابت می ماند و مقیاس محور y پنج برابر می شود و پس از دستور چاپ که در خط ششم قرار دارد کلمه POSTSCRIPT به صورت کشیده چاپ می شود. اگر عملوند این دستور به صورت یک عدد منفی باشد، متن خروجی به صورت معکوس است. نمونه ای از این کاربرد را در مثال زیر می بینید:


کد برنامه	اجرای برنامه
<pre> /Times-Italic findfont 50 scalefont setfont 432 200 translate -1 1 scale 50 50 moveto (Mirror) show </pre>	

در این برنامه می بینیم که پس از دستور خط پنجم جهت محور X تغییر می کند و در خط هفتم متن به صورت آینه در صفحه چاپ می شود.

تا اینجا دستورات مورد بحث در مورد صفحه مختصات و عملگرهای متنی بودند، حال با تعدادی از عملگرهای گرافیکی آشنا می شویم.

### **:Lineto**

این دستور از محل قرار گرفتن نوک قلم در حال حاضر، خطی تا نقطه (x,y) که عملوند این عملگر است کشیده می شود.

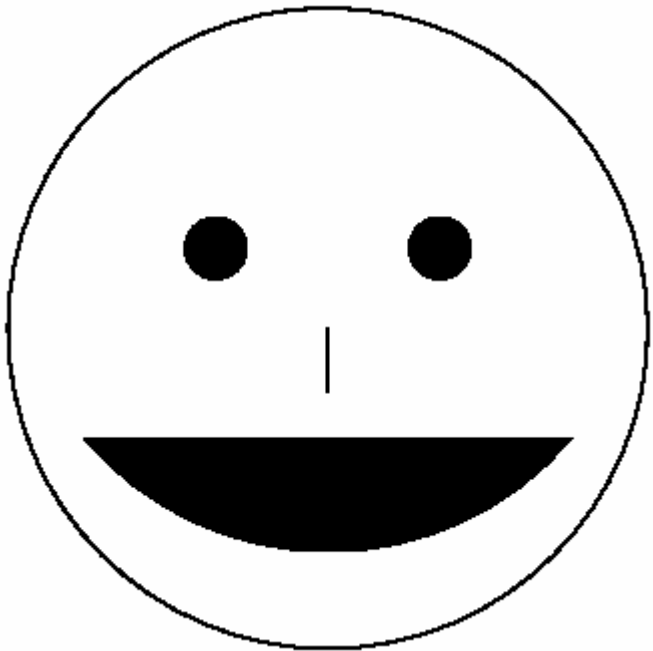
کد برنامه	اجرای برنامه
<pre>newpath 44 800 moveto 44 700 lineto stroke showpage</pre>	

دستور اول مشخص می کند که مسیر جدیدی در حال ایجاد شدن است. دستور دوم و سوم مختصات خطی از نقطه (44,800) تا نقطه (44,700) مشخص می کنند و دستور stroke این مسیر را روی صفحه رسم می کند.

### **:Arc**

این عملگر ۵ عملوند دارد. دو عملوند اول x و y هستند و مرکز کمان را مشخص می کند. عملوند سوم شعاع کمان را تعیین می کند و دو عملوند بعد ابتدا و انتهای کمان را بر حسب درجه مشخص می کنند.

در زیر مثالی از عملگرهای گرافیکی را می بینیم:

کد برنامه	اجرای برنامه
<pre> 300 600 translate newpath 0 0 100 0 360 arc closepath stroke newpath -35 25 10 0 360 arc fill closepath stroke newpath 35 25 10 0 360 arc fill closepath stroke newpath 0 30 100 220 320 arc fill closepath stroke newpath 0 0 moveto 0 -20 lineto stroke showpage </pre>	

### ساختارهای کنترل شرط و تکرار:

زبان پست اسکریپت نیز مانند سایر زبانها دارای ساختارهای کنترل شرط و تکرار است، و تفاوت آن تنها در ظاهر این دستورات است که به علت postfix بودن زبان پست اسکریپت، درک آن کمی برای کاربر مشکل است.

به طور مثال در قطعه کد زیر تابع فیبوناچی را به زبان پست اسکریپت می بینیم. همچنین معادل این کد به زبان C نیز برای مقایسه و درک بهتر درج شده است:

کد فیبوناچی با پست اسکریپت	کد فیبوناچی با C
<pre> /fib { dup 1 eq { } { dup 2 eq { pop 1 } { dup 1 sub fib exch 2 sub fib add } ifelse } ifelse } def </pre>	<pre> int fib (int n) { if (n==1) { return n; } else { if (n==2) { return 1; } else { return fib(n-1)+fib(n-2); } } } </pre>

البته زبان پست اسکریپت به خاطر ماهیت postfix بودنش عملگرهای آن عملوندهای خود را از روی پشته بدست می آورند و نتیجه خود را نیز نهایتاً روی پشته قرار می دهند. قطعه کد به زبان C طوری مرتب شده است که دستورات معادل در زبان پست اسکریپت و C تقریباً مقابل یکدیگر باشند.

در اولین خط از این قطعه کد مشخص می شود که بدنه یک تابع جدید آغاز می شود. در خط دوم عملگر dup عنصر بالای پشته را دوباره در پشته تکرار می کند. بدین ترتیب پس از اجرای خط دوم از عملوند تابع دو مقدار روی پشته قرار دارد. در خط سوم مقدار عملوند از پشته خارج می شود و با عدد ۱ مقاسیه می گردد. چون در خط دوم یک کپی دیگر از عملوند را در پشته قرار داده بودیم، پس از اجرای خط سوم هنوز یک نمونه از عملوند تابع روی پشته موجود است. در صورت برابر بودن عملوند با عدد ۱ تابع خاتمه می یابد. در این صورت چون هنوز یک نمونه از عملوند (که همان عدد ۱ است) روی پشته موجود است و جواب این تابع هم همین ۱ است که روی پشته قرار دارد.


به همین ترتیب در صورت برابر نبودن عدد با ۱، شرط برابری عملوند با عدد ۲ کنترل می شود. در صورت برابر بودن عملوند با عدد ۲، عملوندی که روی پشته قرار دارد از پشته خارج می شود و عدد ۱ را روی پشته قرار می گیرد. در این صورت خروجی تابع همین عدد ۱ می باشد.

در صورتیکه دو شرط قبل برآورده نشدند، تابع بازگشتی فراخوانی می گردد. ابتدا یک کپی از عملوند روی پشته قرار می گیرد و سپس تابع با مقدار  $n-1$  فراخوانی می شود. پس از اجرای این دستور، روی پشته حاصل تابع با ورودی  $n-1$  قرار دارد و عنصر زیر آن یک کپی از عملوند (همان  $n$ ) قرار دارد. لذا برای فراخوانی تابع با ورودی  $n-2$ ، دو عنصر بالای پشته را با هم جابجا می شوند و سپس تابع برای  $n-2$  فراخوانی می شود. سر انجام حاصل جمع دو مقدار  $fib(n-2)$  و  $fib(n-1)$  که اکنون هر دو روی پشته قرار دارند محاسبه می شود و جواب روی پشته قرار می گیرد که همان خروجی تابع است.

در این مثال ملاحظه شد که پست اسکریپت از توابع بازگشتی نیز پشتیبانی می کند که البته این ویژگی به خاطر وجود پشته در این زبان است.

حلقه نیز در زبان پست اسکریپت وجود دارد. در قطعه کد زیر یک نمونه از کاربرد حلقه در این زبان را می

بینیم:

کد برنامه	اجرای برنامه
<pre> /Times-Italic findfont 30 scalefont setfont /printPS { 0 0 moveto (postscript) show } def 320 400 translate .95 -.05 0 {setgray printPS -1 .5 translate } for 1 setgray printPS showpage </pre>	

ابتدا نوع و اندازه فونت مورد نظر انتخاب می گردد. سپس رویه `printPS` تعریف می شود. در دستور بعد مبدا مختصات به نقطه (۳۲۰،۴۰۰) منتقل می گردد. در خط ششم از برنامه سه عدد قرار دارند که به ترتیب ابتدا، `step` حلقه و انتهای شمارنده حلقه هستند.

در خط هفتم دستورات حلقه قرار دارند که به تعداد ۱۹ بار (  $0.95 / 0.05 = 19$  ) تکرار می گردد، که در هر بار اجرای حلقه رنگ نوشتن به اندازه شمارنده حلقه روشن می گردد که وقتی شمارنده برابر ۰٫۹۵ است، رنگ روشن و وقتی برابر ۰ می شود کاملاً تیره می شود. پس از تغییر رنگ قلم نوشتن، مبدا کمی جابجا می گردد و رویه `printPS` فراخوانی می شود. پس از پایان حلقه قلم کاملاً روشن می گردد و دوباره رویه `printPS` فراخوانی می گردد.

### جمع بندی:

پست اسکریپت زبانی است که قابلیت‌های زیادی را در اختیار کاربر قرار می دهد. با این وجود، به دو دلیل کار کردن با آن کمی سخت است: اول اینکه این زبان `postfix` است و همین باعث ایجاد پیچیدگی در ذهن کاربر می شود. از طرف دیگر عدم پشتیبانی مستقیم از `Unicode` نیز کاربرانی را که از زبانی غیر از زبان انگلیسی استفاده می کنند با مشکلاتی مواجه می کند. اما وقتی قرار باشد از نرم افزارهایی جهت تبدیل فرمت اولیه به پست اسکریپت استفاده شود (نظیر پریترهای مجازی که خروجی آنها پست اسکریپت است) کاربر از این مشکلات دور نگه داشته می شود و پست اسکریپت به یکی از فرمت‌های متداول برای انتقال اطلاعات تبدیل می شود.

منابع:

**Postscript Language Reference Manual** (Adobe Systems Inc)

**Postscript Language Tutorial & Cookbook** (Adobe Systems Inc)

<http://www.mefco.com/psc/index.html>