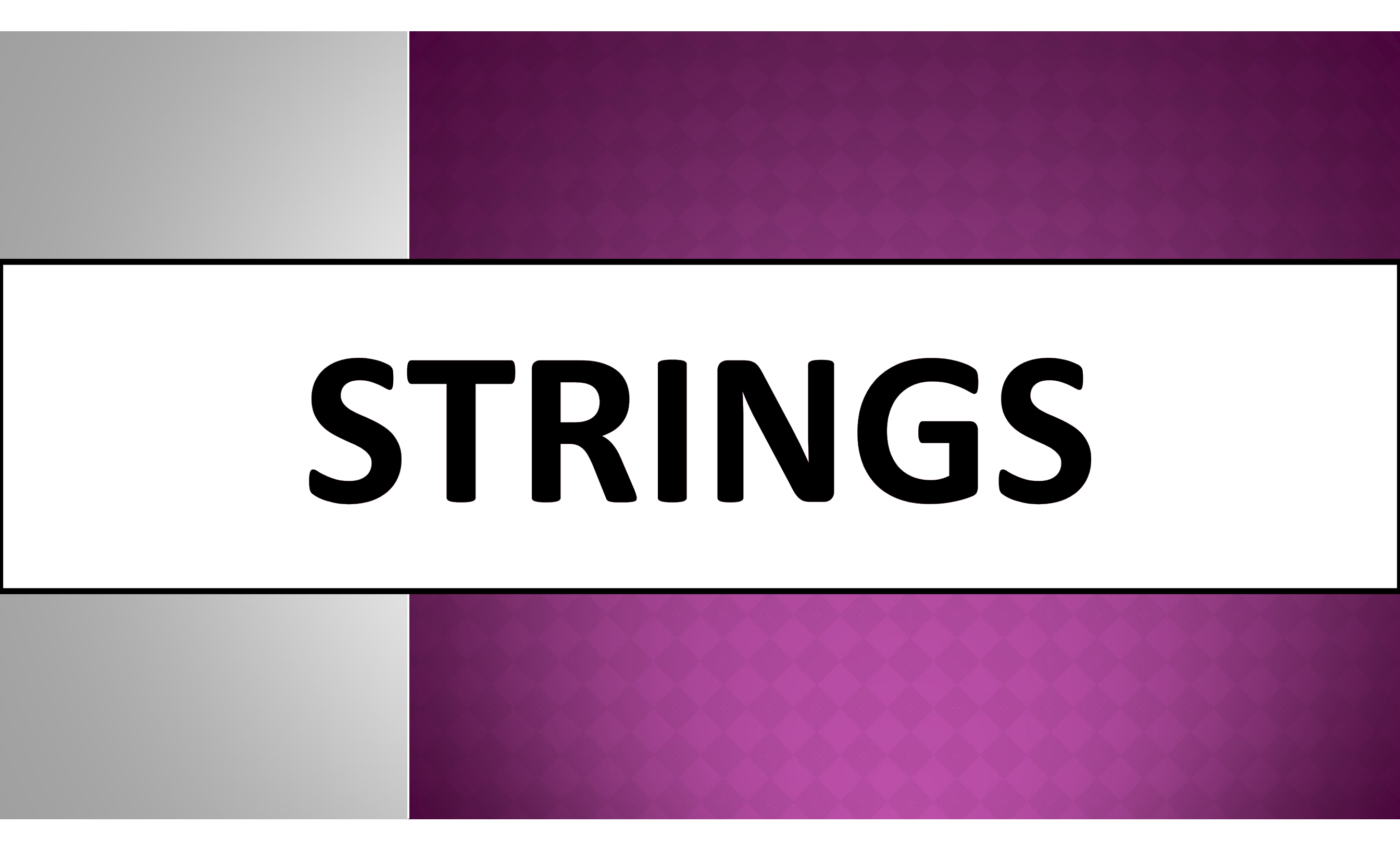


The image features a central white rectangular area containing the word "STRINGS" in a large, bold, black sans-serif font. This central area is flanked on both the left and right sides by vertical panels. The left panels are a solid light grey color, while the right panels are a dark purple color with a subtle, repeating diamond-shaped pattern. The overall layout is clean and modern, with a strong contrast between the black text and the white background.

STRINGS

What are Strings

- Character arrays may also called strings.
- Character arrays or strings are used by programming languages to manipulate text such as words and sentences.
- A string constant is a one-dimensional array of characters terminated by a null (`'\0'`).

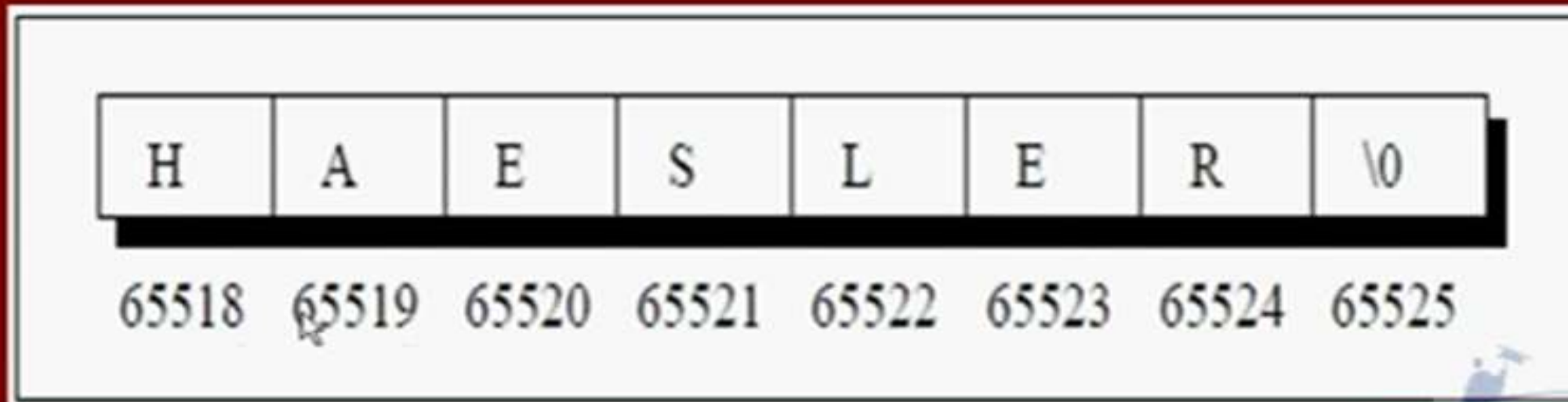
Ex:

```
char name[ ] = { 'H', 'A', 'E', 'S', 'L', 'E', 'R', '\0' } ;
```

- Each character in the array occupies one byte of memory and the last character is always `'\0'`.
- `'\0'` is called null character.
- Note that `'\0'` and `'0'` are not same. ASCII value of `'\0'` is 0, whereas ASCII value of `'0'` is 48.

What are Strings

- Note that the elements of the character array are stored in contiguous memory locations.
- The terminating null ('\0') is important, because it is the only way the functions that work with a string can know where the string ends. In fact, a string not terminated by a '\0' is not really a string, but merely a collection of characters.



Initializing strings

- ⦿ **char c[] = "abcd";**
- ⦿ **char c[50] = "abcd";**
- ⦿ **char c[] = {'a', 'b', 'c', 'd', '\0'};**
- ⦿ **char c[5] = {'a', 'b', 'c', 'd', '\0'};**

Reading a string

```
#include<iostream.h>
#include<conio.h>
void main()
{
clrscr();
char name[20];
cout<<"Enter name: ";
cin>>name;
cout<<"Your name is: "<<name;
getch();
}
```

Example - String

```
void main()
{
char name[]="hello";
int i=0;
while((i<6))
{
cout<<name[i];
i++;
}
}
```

```
void main()
{
char name[10];
cout<<"ENTER YOU NAME : ";
for(int i=0;i<10;i++)
{
cin>>name[i];
}
cout<<"YOUR NAME : ";
for(i=0;i<10;i++)
{
cout<<name[i];
}
}
```

Example : fgets() and puts()

```
int main()
{
clrscr();
char name[30];
cout<<"Enter name:";
fgets(name,sizeof(name),stdin);
cout<<"Name:";
puts(name);
getch();
}
```

You can use the fgets() function to read a line of string. And, you can use puts() to display the string.

Passing Strings to Functions

Passing Strings to Functions

```
#include<stdio.h>
void DisplayString(char str[]);
void main()
{
char str[50];
cout<<"Enter string:";
fgets(str,sizeof(str),stdin);
DisplayString(str);
}
```

```
void DisplayString(char str[])
{
cout<<"String Output:";
puts(str);
}
```

Strings and Pointers

```
void main(void)
{
char name[]="Harry Potter";
cout<<*name;
cout<<*(name+1);
cout<<*(name+6);
cout<<*(name+7);
}
```

Standard Library

String Functions

Operations on Strings

- Reading and displaying strings.
- Copying one string to another.
- Combining strings.
- Comparing strings.
- Extraction of a portion of a strings.
- checking string for equality.

- strcpy : Copies a string into another.
- strncpy : Copies first n characters of one string into another.
- strcmp : Compares two strings.
- strncmp : Compares first n characters of two strings.
- strcmpi : Compares two strings without regard to case ("i" denotes that this function ignores case).
- stricmp : Compares two strings without regard to case (identical to strcmpi).
- strncmpi : Compares first n characters of two strings without regard to case.

Example: C strlen() function

```
#include<iostream.h>
#include<string.h>
#include<conio.h>
void main()
{
clrscr();
char a[20]="program";
char b[20]={'p','r','o','g','r','a','m','\0'};
char c[20]="viva Technologies";
cout<<"Length of string a=\n"<<strlen(a)<<endl;
cout<<"Length of string b=\n"<<strlen(b)<<endl;
cout<<"Length of string c=\n"<<strlen(c)<<endl;
getch();
}
```

Example: C strlen() function

```
#include<iostream.h>
#include<string.h>
#include<conio.h>
void main()
{
clrscr();
    char arr[]="happy";
    int len1,len2;
    len1=strlen(arr);
    len2=strlen("school");
    cout<<"\n string ="<<arr<<"\t length = " <<len1;
    cout<<"\n string = "<<"school "<<"\t length = "<<len2;
    getch();
}
```

Example: C strcpy()

```
#include <stdio.h>
```

```
#include <string.h>
```

```
void main()
```

```
{ char str1[20] = "Viva Technologies";
```

```
char str2[20];
```

```
strcpy(str2, str1); // copying str1 to str2
```

```
puts(str2); // Printing Viva Technologies
```

```
getch();
```

```
}
```

Example: C strcpy()

```
#include<iostream.h>
#include<string.h>
#include<conio.h>
#include<stdio.h>
void main()
{
char str1[20]="viva Technologies";
char str2[20];
strcpy(str2,str1);
puts(str2);
}
```

Example: C strcmp() function

```
#include<iostream.h>
#include<string.h>
#include<conio.h>
void main()
{
clrscr();
char str1[]="Berry",str2[]="Ferry",str3[]="Jerry";
int result;
result=strcmp(str1,str2);
cout<<"strcmp(str1,str2)= "<<result<<endl;
result=strcmp(str1,str3);
cout<<"strcmp(str1,str3)= "<<result<<endl;
getch();}
```

Example: C strcat() function

```
#include <stdio.h>
```

```
#include <string.h>
```

```
int main()
```

```
{
```

```
char str1[100] = "This is ", str2[ ] = "Viva Technologies";
```

```
strcat(str1, str2); // concatenates str1 and str2
```

```
puts(str1);
```

```
puts(str2);
```

```
return 0;
```

```
}
```

```
// the resultant string is stored in str1.
```

Example: C strcat() function

```
#include<iostream.h>
#include<string.h>
#include<conio.h>
void main()
{
clrscr();
char source[]="Folks!";
char target[30]="Talk!";
strcat(target ,source);
cout<<"\n source string = "<<source;
cout<<"\n after concatination string ="<<target;
getch();
}
```

Example: C strcat() function

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
void main()
{
clrscr();
char str1[100]=" This is ",str2[]=" viva Technologies ";
strcat(str1,str2);
puts(str1);
puts(str2);
getch();
}
```

THANKS