

`FILE *fopen(char *filename, char *mode)`

Open een bestand (H4, slide 8)

`int fclose(FILE *fp)`

Sluit een bestand (H4, slide 9)

`int fgetc(FILE *fp)`

Lees karakter in uit bestand (H4, slide 9)

`int fputc(char c, FILE *fp)`

Schrijf karakter naar bestand (H4, slide 9)

`char *fgets(char *line, int maxline, FILE *fp);`

Lees lijn in (H4, slide 19)

`int fputs(char *string, FILE *fp)`

Schrijf string naar bestand (H4, slide 19)

`int fprintf/fscanf(FILE *fp, char *format, ...)`

Schrijf/lees formaat naar/uit bestand (H4, slide 14)

```
FILE *fopen(char *filename, char *mode)
```

Open een bestand (H4, slide 9)

```
FILE *file = fopen("bestand.txt", "r");
```

```
FILE *file = fopen("/pad/naar/bestand/bestand.txt", "w");
```

```
int fclose(FILE *fp)
```

Sluit een bestand (H4, slide 10)

```
fclose(file);
```

A screenshot of a text editor window titled "bestand.txt". The window contains 15 lines of text, each starting with "Dit is lijn " followed by a number from 1 to 15, and ending with a period. The text is displayed in a monospaced font.

```
FILE *file = fopen("bestand.txt", "r");
```

```
int fgetc(FILE *fp)
```

Lees karakter in uit bestand (H4, slide 10)

```
char c = fgetc(file); // 'D'
```

```
fgetc(file); // 'i'
```

```
fgetc(file); // 't'
```

```
fgetc(file); // ' '
```

...

```
fgetc(file); // EOF
```

```
FILE *file = fopen("bestand.txt", "w");
```

```
int fputc(char c, FILE *fp)
```

Schrijf karakter naar bestand (H4, slide 10)

Returnt EOF als er een fout is opgetreden

```
fputc('A', file); // Schrijf 'A'
```

```
fputc('b', file); // Schrijf 'b'
```

A screenshot of a text editor window titled "bestand.txt". The window contains 15 lines of text, each starting with "Dit is lijn " followed by a number from 1 to 15, and ending with a period. The text is displayed in a monospaced font with syntax highlighting: "Dit" is blue, "is" is green, "lijn" is red, and the numbers are black. The cursor is at the end of the 15th line.

```
FILE *file = fopen("bestand.txt", "r");
```

```
char *fgets(char *line, int maxline, FILE *fp);
```

Lees lijn in (H4, slide 19)

Returnt NULL als file volledig is ingelezen

```
char line[1000];
```

```
fgets(line, 1000, file); // line == "Dit is lijn 1."
```

```
FILE *file = fopen("bestand.txt", "w");
```

```
int fputs(char *str, FILE *fp)
```

Schrijf string naar bestand (H4, slide 19)

Returnt EOF als er een fout is opgetreden

```
fputs("Een string", file); // Schrijf "Een string"
```

A screenshot of a text editor window titled "bestand.txt". The window contains 15 lines of text, each starting with "Dit is lijn " followed by a number from 1 to 15, and ending with a period. The text is displayed in a monospaced font.

```
FILE *file = fopen("bestand.txt", "w");
```

```
int fprintf(FILE *fp, char *format, ...);  
    Schrijf formaat naar bestand (H4, slide 15)  
    Retournt aantal karakters weggeschreven
```

```
fprintf(file, "%s %s %i\n", "Dit is", "lijn", 1);  
// Schrijf "Dit is lijn 1"
```

```
int fscanf(FILE *fp, char *format, ...)  
    Lees formaat uit bestand (H4, slide 15)  
    Retournt aantal argumenten dat gematcht werd
```

```
char str1[100], str2[100], str3[100];  
int i;  
fscanf(file, "%s %s %s %i", str1, str2, str3, &i);  
// str1 == "Dit", str2 == "is", str3 == "lijn", i == 1
```

```
fscanf(file, "%i", &i);  
"Dit is lijn 1." kan niet gematcht worden met een integer ->  
fscanf retournt 0
```